Hatch Tutors Web Application:
Combating Education Inequity with Free Online Tutoring

Angela Li
Advisor: Prof. Robert Dondero
May 3, 2021

Submitted in partial fulfillment of the requirements for the Princeton University Computer Science Senior Independent Work. I pledge my Honor that this paper represents my own work in accordance with University regulations.

Table of Contents

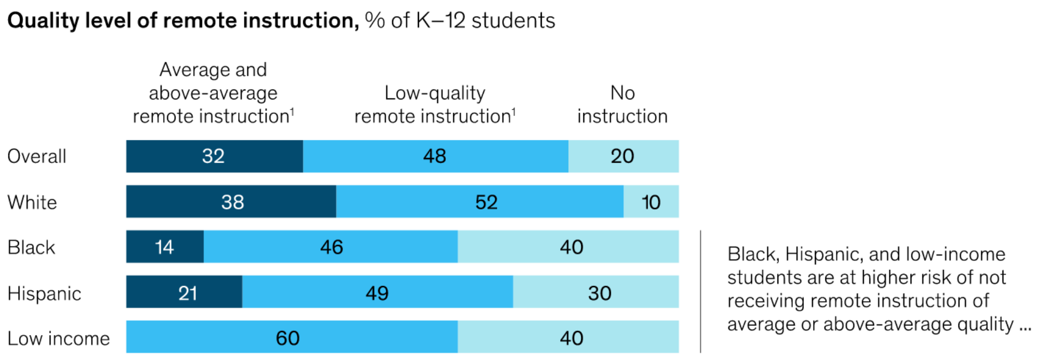
1. Introduction.

In the spring of 2020, the United States began its first lockdown due to the COVID-19 pandemic. The majority of K-12 schools had to shut down their in-person operations, and about 93% of households with school-aged children reported some form of distance learning, which includes "remote" learning where classes are taught synchronously online or "hybrid" learning where students go to school in-person half of the week and participate in remote learning the other half. By June 2020, around 94% of public school teachers reported moving their teaching online.[1] The transition of schools to remote learning in spring of 2020 due to the COVID-19 pandemic presented challenges for K-12 students everywhere in the United States. However, the quality of online classroom learning during school closures varies significantly determined by factors such as access to technology, the quality of remote instruction, and home support for students. A study conducted by Mckinsey in June 2020 predicted learning loss to be greatest in low-income, black, and Hispanic students. In particular, learning loss is exacerbated for low-income students because they are "less likely to have access to high-quality remote learning or to a conducive learning environment, such as a quiet space with minimal distractions, devices they do not need to share, high-speed internet, and parental academic supervision."[2] Data showed that around 60% low-income students surveyed would receive low-quality remote instruction and the other 40% received no remote instruction (Fig. 1). This is a drastic difference to the quality of remote instruction for students from higher-income families.

[1] Bureau, U.S. Census. "Schooling During the COVID-19 Pandemic." *The United States Census Bureau*, 26 Aug. 2020, www.census.gov/library/stories/2020/08/schooling-during-the-covid-19-pandemic.html.

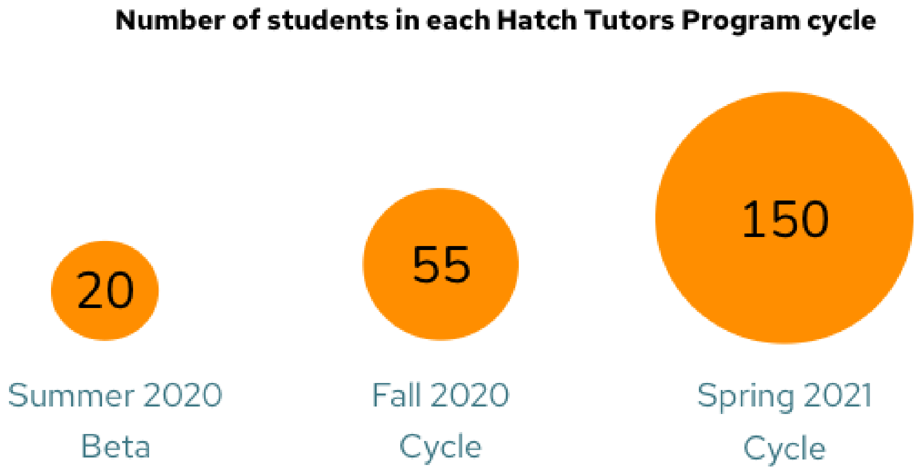
[2] Dorn, Emma, et al. "COVID-19 and Student Learning in the United States: The Hurt Could Last a Lifetime." *McKinsey & Company*, McKinsey & Company, 14 Dec. 2020, www.mckinsey.com/industries/public-and-social-sector/our-insights/covid-19-and-student-learning-in-the-united-states-the-hurt-could-last-a-lifetime#.

FIGURE 1.


Quality level of remote instruction, % of K–12 students

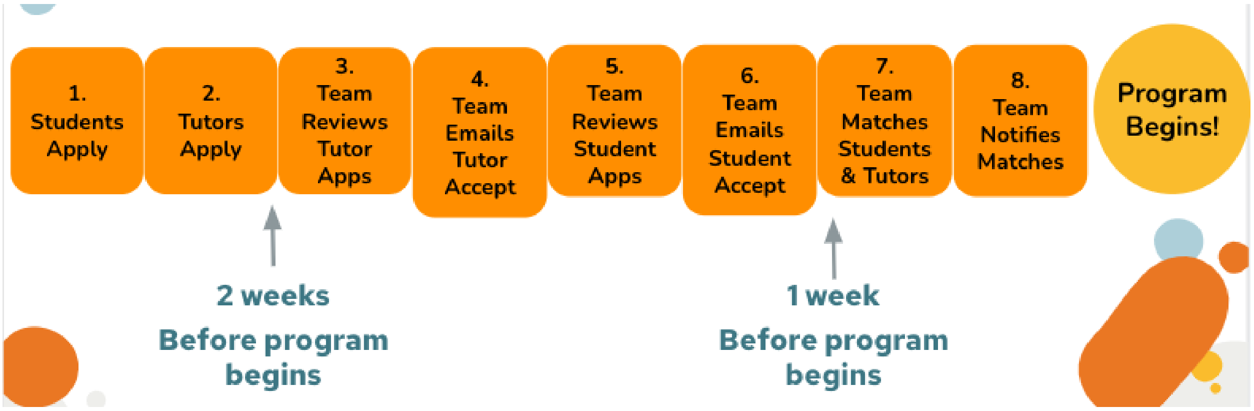
Black, Hispanic, and low-income students are at higher risk of not receiving remote instruction of average or above-average quality ...

To combat the widening gap in education caused by the pandemic, I founded a nonprofit called Hatch Tutors in 2020. Hatch Tutor’s mission is to provide academic support to K-12 students from underserved backgrounds through free online tutoring. Hatch runs a semester-long program where K-12 students are paired with a university-aged tutor for individual tutoring. Students receive tutoring at least once a week in a subject area of their choice, and they use online conferencing platforms such as Zoom to conduct the tutoring session. To date, Hatch Tutors has run a Beta free tutoring program during the summer of 2020 and two more official semester-long program cycles during the Fall of 2020 and the Spring of 2021. Hatch has been steadily growing the number of students in the tutoring program, from 20 students in the beta program to 150 students in the Spring 2021 program where we are today.



Hatch's rapid scaling of the tutoring program is a crucial motivation for my independent work project as I will explain here. At the beginning of every program cycle, the Hatch Executive team spends two weeks doing an intensive application review and manual matching process to pair students with tutors (fig. 2).

FIGURE 2


1. Students Apply
2. Tutors Apply
3. Team Reviews Tutor Apps
4. Team Emails Tutor Accept
5. Team Reviews Student Apps
6. Team Emails Student Accept
7. Team Matches Students & Tutors
8. Team Notifies Matches
Program Begins!
2 weeks
Before program begins
1 week
Before program begins

The process is as follows:

1. K-12 students across the nation apply to the Hatch Tutors program through a Google form application.
2. University-aged tutors apply to the Hatch Tutors program through a Google form application.
3. The Hatch Executive team reviews all of the tutor applications by reading every Google form response.

4. The Hatch Executive team manually emails out acceptance and rejection emails to every tutor applicant.
5. The Hatch Executive team reviews all of the K-12 student applications by reading every Google form response.
6. The Hatch Executive team manually emails out acceptance and rejection emails to every student applicant.
7. The Hatch Executive team manually pairs K-12 students with a tutor by utilizing a spreadsheet with every student and tutors information. Executive team members will ensure that the subject area, grade level, and availability of the student and tutor match.
8. The Hatch Executive team notifies each student-tutor pair of their match by manually sending an email to each individual.

Since all parts of the review process are manual, including the match creation portion, it takes a tremendous amount of time and effort to complete. We as a team spent likely around 100 hours to complete the process for 150 students and 150 tutors before the Spring 2021 program. With limited people on the executive team, this process is highly inefficient and unscalable and as Hatch continues to expand its reach to support more K-12 students, the matching will only become more complicated. This is the motivation behind my independ work project. My goal in creating the Hatch Tutors Web Application is to automate the exact process I delineated in order to remove a manual bottleneck for the Executive team and allow the tutoring program to scale to more K-12 students nationwide.

2. Background and Related Work.

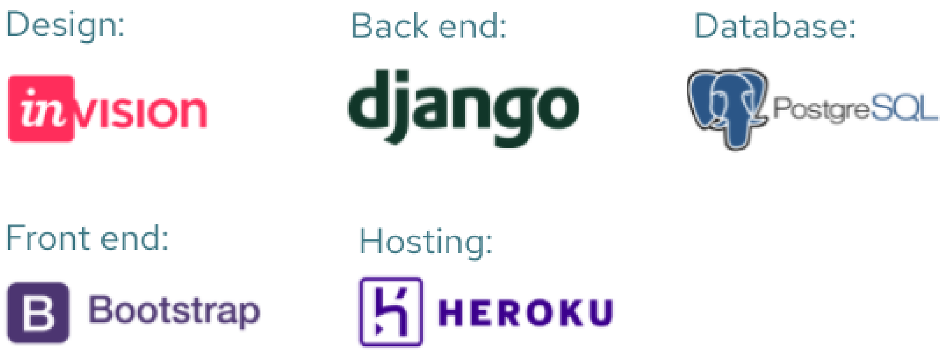
Since this independent work project is a web application being built from scratch and not a theoretical project, there is not much related work of relevance. There does exist a current website for Hatch Tutors at the domain hatchtutors.com which is a completely static website created from Squarespace’s drag-and-drop templates. The Squarespace website has limited customization ability, does not have a backend, and requires that I pay a monthly fee to maintain its existence. For these reasons, I have decided to code my own web application to replace the

Squarespace that will include all the functionality needed to automate the program application review process.

3. Approach

a. Tools

I chose the following tools in developing the Hatch Tutors web application:



I will briefly describe the reasons why I chose these specific tools for web application development.

1. inVision is a popular digital product design platform that is often used to create mockups for web application user interfaces. My colleague in Hatch and UX / UI designer, Theresa Salud, is familiar with inVision and was able to guide me through the usage.
2. Django is a popular web framework and I have previously used it to develop a web application in COS 333.
3. PostgreSQL is a relational database often used with Django.
4. Bootstrap is a popular CSS framework that I chose to work with because it has broad open-source resources online including pre-made aesthetic front end templates. I also considered using React for the front end but chose Bootstrap because of familiarity and ease of usage.

5. I chose Heroku to host the web application because Heroku provides a free 10K database rows that I can use during development and because I am familiar with it from COS 333.

b. Development Approach

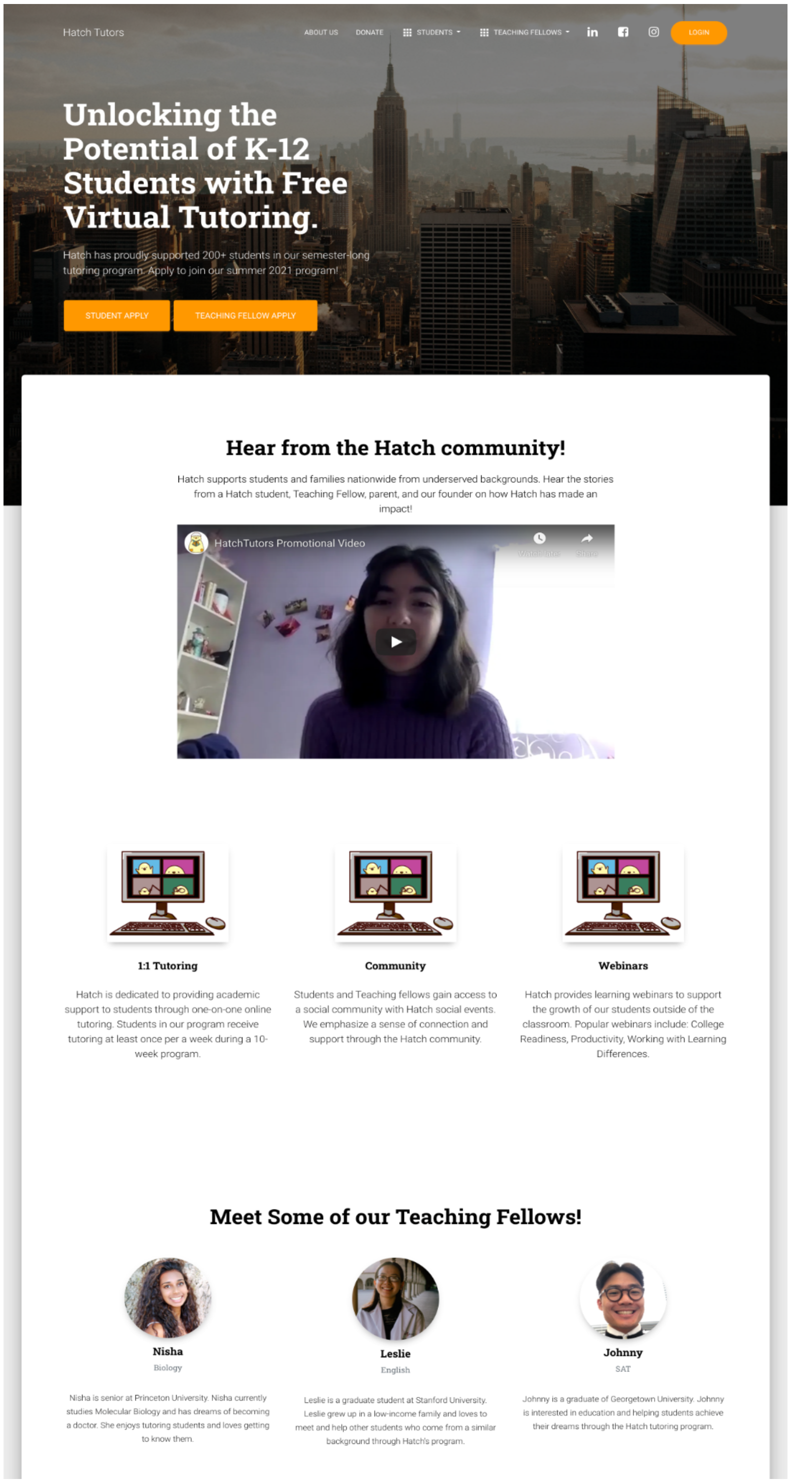
My general approach to development was the following:

1. I created low-fidelity mockups on paper of the potential user interface for the application. Then, I tested the mockups by interviewing Hatch Executive team members and asking them to trace through the mockups and provide feedback on how the proposed features reflect the real-life application review process. I continued to iterate on the mockups based on team feedback and A/B testing versions in collaboration with my colleague Theresa. Then, I created high-fidelity mockups of the web application user interface using inVision that I could refer to during the development process. Refer to APPENDIX 1 for the mockups.
2. I implemented the web application step by step in the order of the application review and matching process from Fig. 2:
    a. I first implemented the homepage and the data collection portion, which includes the student and tutor application forms.
    b. Then I implemented the application review portals for students and tutors. I included email notification functionality.
    c. Finally, I implemented the student-tutor matching portal.

4. Implementation - Functionality

In this section, I will discuss the use case and functionality of each page in the web application. I will then map each page to the steps of the application review and matching process in Fig. 2.

Homepage



Unlocking the Potential of K-12 Students with Free Virtual Tutoring.

Hatch has proudly supported 200+ students in our semester-long tutoring program. Apply to join our summer 2021 program!

Hear from the Hatch community!

Hatch supports students and families nationwide from underserved backgrounds. Hear the stories from a Hatch student, Teaching Fellow, parent, and our founder on how Hatch has made an impact!

1:1 Tutoring

Hatch is dedicated to providing academic support to students through one-on-one online tutoring. Students in our program receive tutoring at least once per a week during a 10-week program.

Community

Students and Teaching fellows gain access to a social community with Hatch social events. We emphasize a sense of connection and support through the Hatch community.

Webinars

Hatch provides learning webinars to support the growth of our students outside of the classroom. Popular webinars include: College Readiness, Productivity, Working with Learning Differences.

Meet Some of our Teaching Fellows!

Nisha

Biology

Nisha is senior at Princeton University. Nisha currently studies Molecular Biology and has dreams of becoming a doctor. She enjoys tutoring students and loves getting to know them.

Leslie

English

Leslie is a graduate student at Stanford University. Leslie grew up in a low-income family and loves to meet and help other students who come from a similar background through Hatch's program.

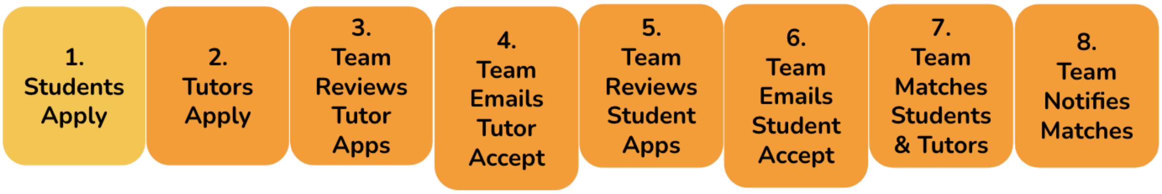
Johnny

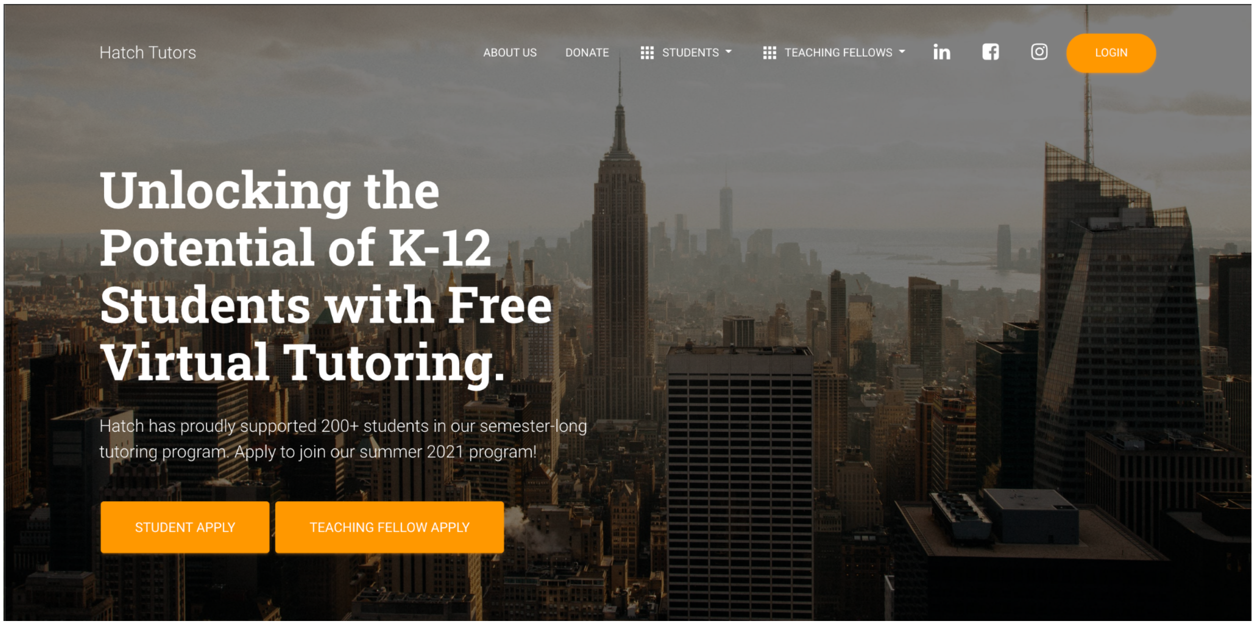
SAT

Johnny is a graduate of Georgetown University. Johnny is interested in education and helping students achieve their dreams through the Hatch tutoring program.

The homepage is the first web page that users will see when they navigate to the Hatch Tutors website. The homepage is meant to give general users an overview of the Hatch Tutors organization. The general sections of the homepage from top to bottom include: 1) the mission statement at the very top plus buttons leading to the student application and tutor application, 2) a video and explanation of the Hatch program and community, 3) a selection of tutor profiles. Now, I will begin giving a step-by-step mapping of the process to the web pages.

STEP 1: Students Apply


1. Students Apply
2. Tutors Apply
3. Team Reviews Tutor Apps
4. Team Emails Tutor Accept
5. Team Reviews Student Apps
6. Team Emails Student Accept
7. Team Matches Students & Tutors
8. Team Notifies Matches

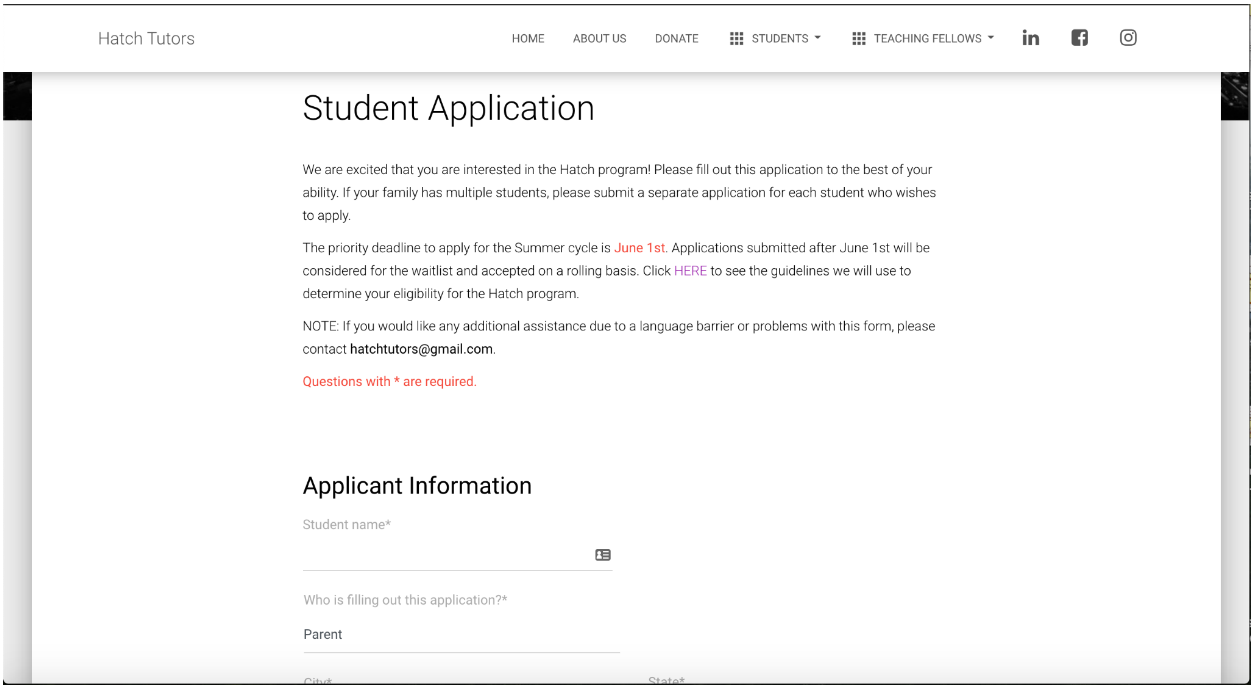


Unlocking the Potential of K-12 Students with Free Virtual Tutoring.

Hatch has proudly supported 200+ students in our semester-long tutoring program. Apply to join our summer 2021 program!

I will walk through this section as a user who is a K-12 student seeking to participate in the Hatch tutoring program. First, I navigate to the homepage for Hatch Tutors and I see a large orange button on the bottom left that says “Student Apply.”

I click on the “Student Apply” button which brings me to this page titled “Student Application.”



Student Application

We are excited that you are interested in the Hatch program! Please fill out this application to the best of your ability. If your family has multiple students, please submit a separate application for each student who wishes to apply.

The priority deadline to apply for the Summer cycle is June 1st. Applications submitted after June 1st will be considered for the waitlist and accepted on a rolling basis. Click HERE to see the guidelines we will use to determine your eligibility for the Hatch program.

NOTE: If you would like any additional assistance due to a language barrier or problems with this form, please contact hatchtutors@gmail.com.

Questions with * are required.

Applicant Information

Student name*

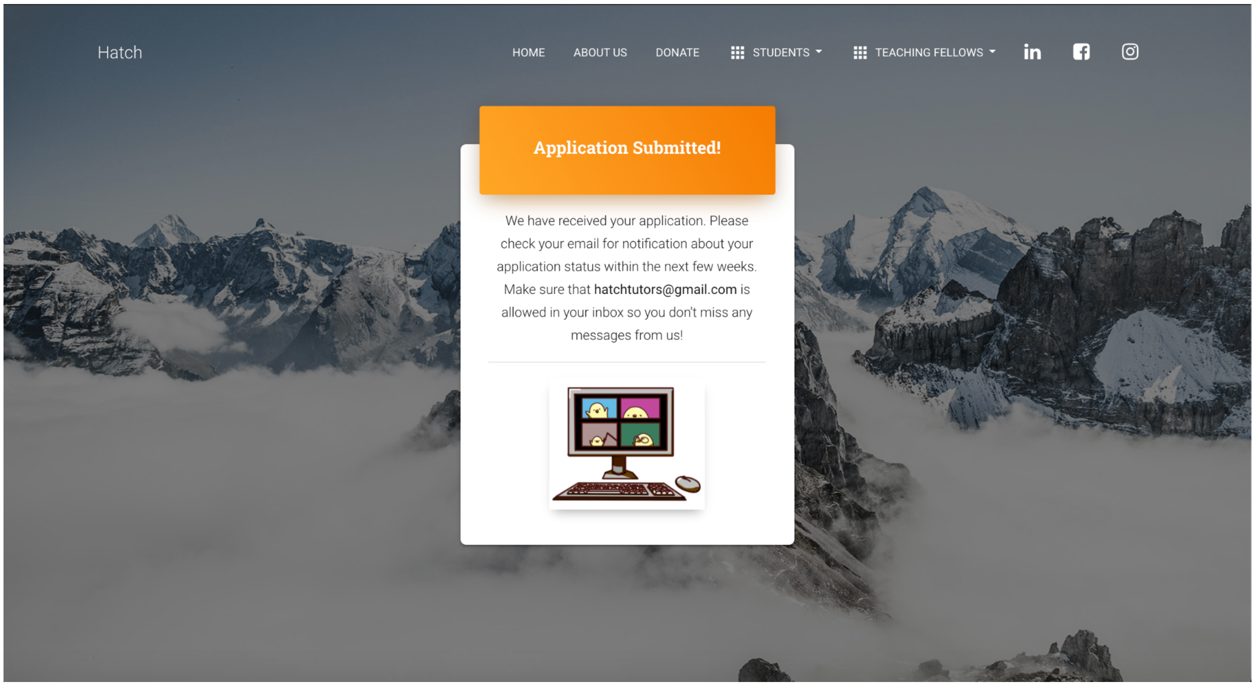
Who is filling out this application?*

Parent

City*

State*

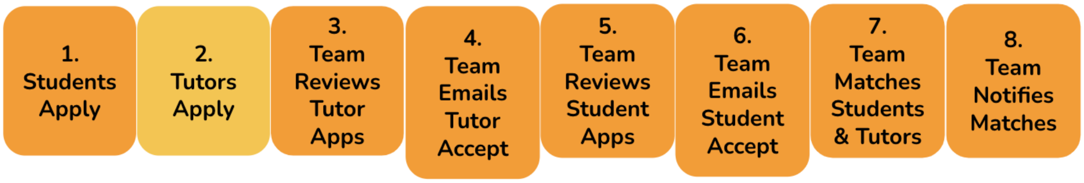
The Student Application page includes instructions at the top that tell me the deadline for submitting the application, links to rules that determine my eligibility for the program (ref. FUTURE WORK), and information about assistance filling out the application. I scroll down and directly fill in the application form with my answers. The large titled sections of the application include: Applicant Information, Caregiver Information, Family Information, other Student Information, 3 choices of Tutoring Subjects, and Tutoring Session Logistics such as availability and access to technology. At the bottom of the application form there is a Submit button. I click the submit button after filling out my information, and then I am brought to this application submission confirmation page. It tells me to check my email for notifications from hatchtutors@gmail.com in the next two weeks.



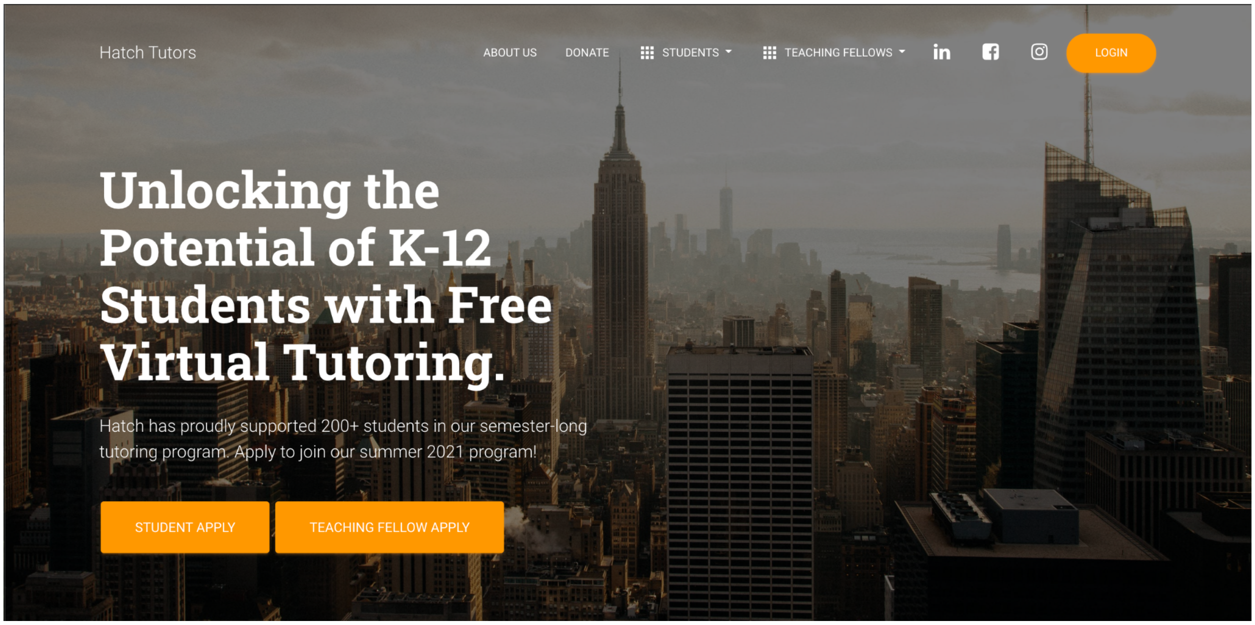
Application Submitted!

We have received your application. Please check your email for notification about your application status within the next few weeks. Make sure that hatchtutors@gmail.com is allowed in your inbox so you don't miss any messages from us!

STEP 2: Tutors Apply



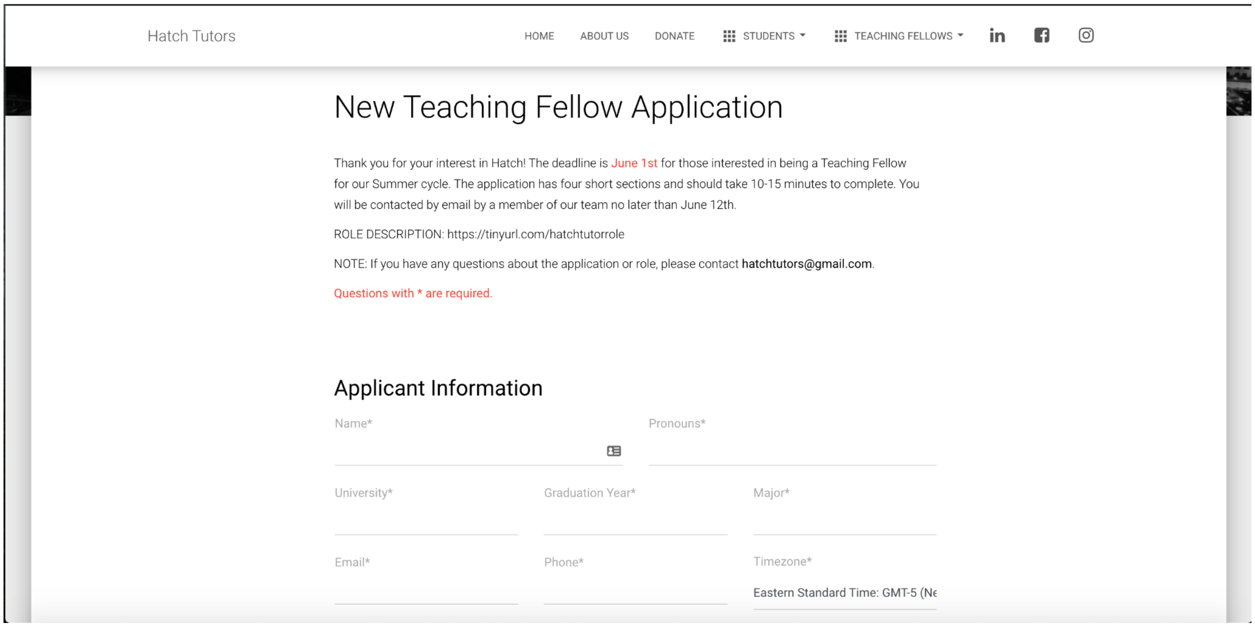
1. Students Apply
2. Tutors Apply
3. Team Reviews Tutor Apps
4. Team Emails Tutor Accept
5. Team Reviews Student Apps
6. Team Emails Student Accept
7. Team Matches Students & Tutors
8. Team Notifies Matches



Unlocking the Potential of K-12 Students with Free Virtual Tutoring.

Hatch has proudly supported 200+ students in our semester-long tutoring program. Apply to join our summer 2021 program!

I will walk through this section as a user who is a university-aged individual seeking to be a tutor in the Hatch tutoring program. First, I navigate to the homepage for Hatch Tutors and I see a large orange button on the bottom left that says “Teaching Fellow Apply.” I click on the “Teaching Fellow Apply” button which brings me to this new page titled “New Teaching Fellow Application.”



New Teaching Fellow Application

Thank you for your interest in Hatch! The deadline is June 1st for those interested in being a Teaching Fellow for our Summer cycle. The application has four short sections and should take 10-15 minutes to complete. You will be contacted by email by a member of our team no later than June 12th.

ROLE DESCRIPTION: https://tinyurl.com/hatchtutorrole

NOTE: If you have any questions about the application or role, please contact hatchtutors@gmail.com.

Questions with * are required.

Applicant Information

Name*

Pronouns*

University*

Graduation Year*

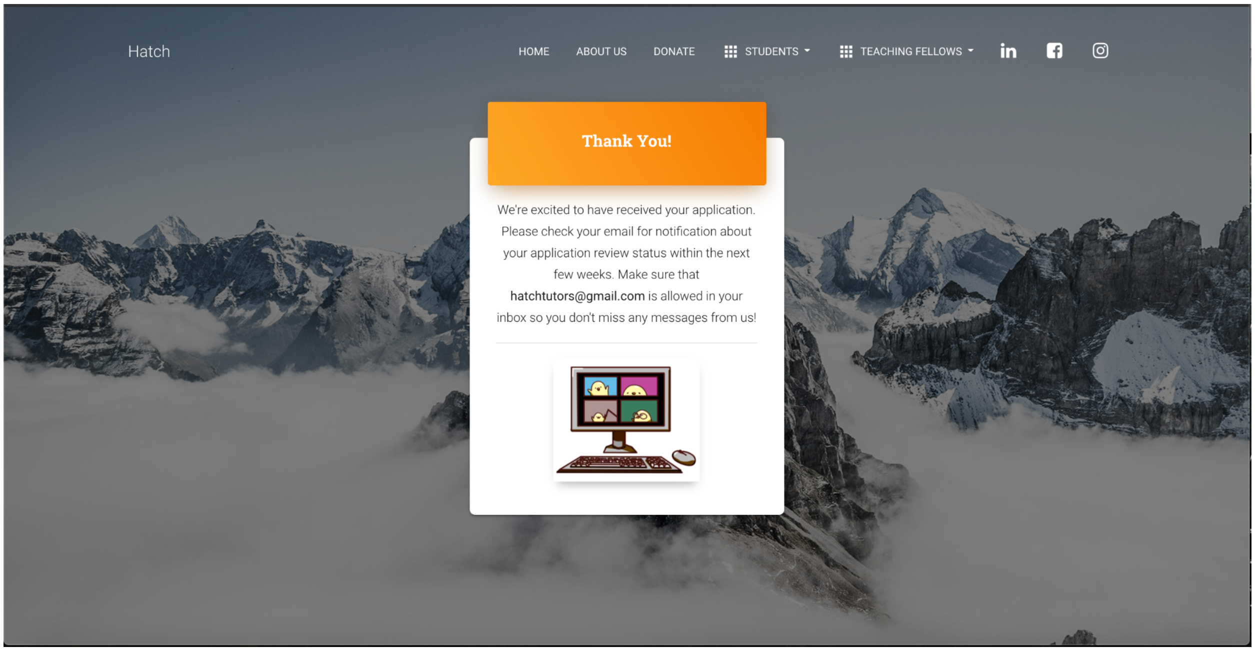
Major*

Email*

Phone*

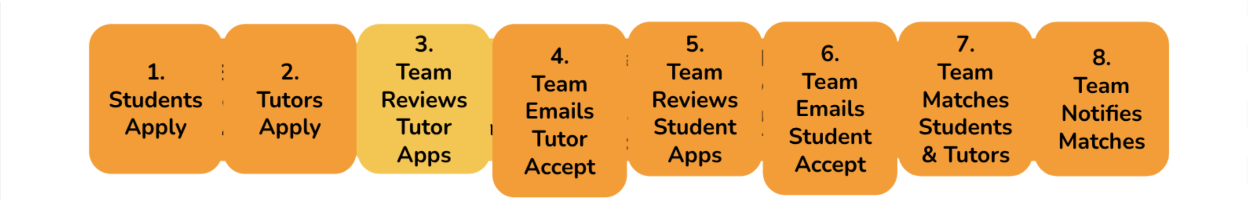
Timezone*
Eastern Standard Time: GMT-5 (Ne

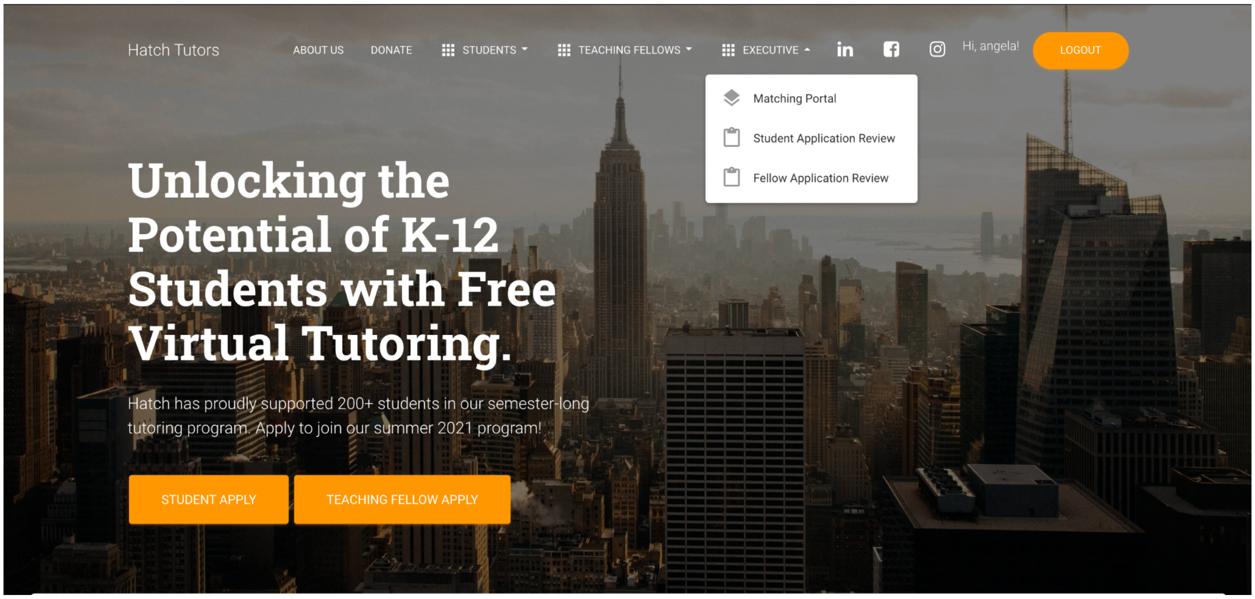
At the top of the New Teaching Fellow Application, there are instructions that tell me the deadline for submitting the application, a link to the tutor role description, and information about how to get help filling out the application. I scroll down and directly fill in the application form with my answers. The large titled sections of the application include: Applicant Information, Short Answer Questions that assess my fit with the program, and Tutoring Logistics. At the bottom of the application form there is a Submit button. I click the submit button after filling out my information, and then I am brought to this application submission confirmation page. It tells me to check my email for notifications from hatchtutors@gmail.com in the next two weeks.



We're excited to have received your application. Please check your email for notification about your application review status within the next few weeks. Make sure that hatchtutors@gmail.com is allowed in your inbox so you don't miss any messages from us!

STEP 3: Team Reviews Tutor Applications

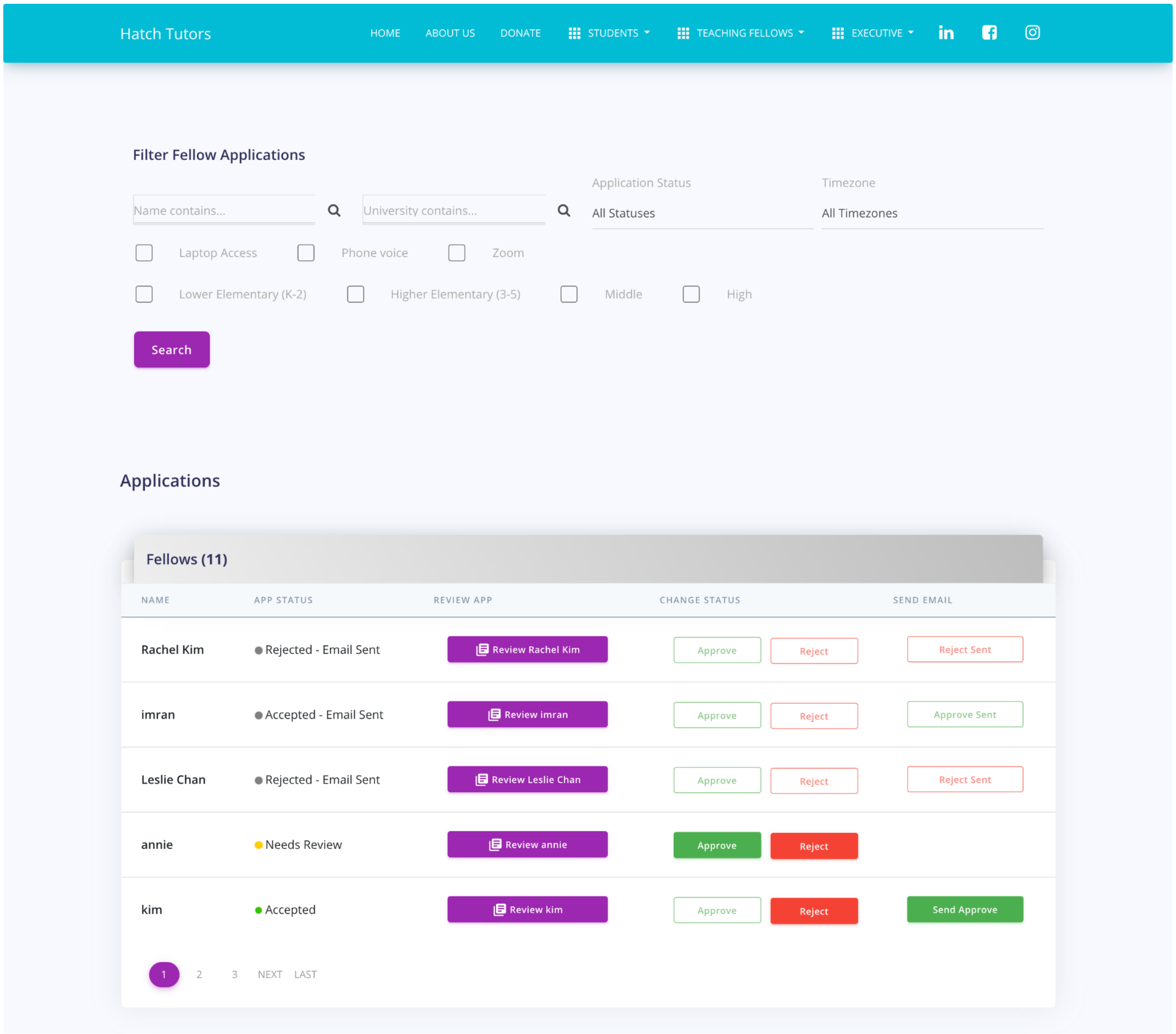

1. Students Apply
2. Tutors Apply
3. Team Reviews Tutor Apps
4. Team Emails Tutor Accept
5. Team Reviews Student Apps
6. Team Emails Student Accept
7. Team Matches Students & Tutors
8. Team Notifies Matches



Unlocking the Potential of K-12 Students with Free Virtual Tutoring.

Hatch has proudly supported 200+ students in our semester-long tutoring program. Apply to join our summer 2021 program!

I will walk through this section as a Hatch Executive team member who is reviewing tutor applications two weeks before the tutoring program begins. First, I navigate to the homepage for Hatch Tutors and I click the login button at the top right which then allows me to login using Google authentication. When I am logged in, I am authenticated as an Executive team member and the navigation bar on the homepage shows a dropdown labeled “Executive.” I click on “Fellow Application Review” from the “Executive” dropdown. This redirects me to a portal where I can review tutor applications.



Filter Fellow Applications

Name contains...

University contains...

Application Status: All Statuses

Timezone: All Timezones

□ Laptop Access □ Phone voice □ Zoom

□ Lower Elementary (K-2) □ Higher Elementary (3-5) □ Middle □ High

Applications

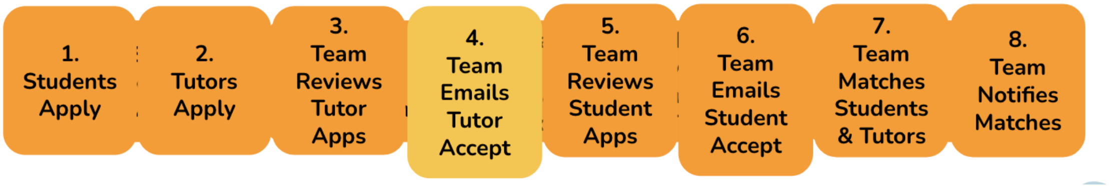
Fellows (11)

NAME	APP STATUS	REVIEW APP	CHANGE STATUS	SEND EMAIL
Rachel Kim	● Rejected - Email Sent	Review Rachel Kim	Approve Reject	Reject Sent
imran	● Accepted - Email Sent	Review imran	Approve Reject	Approve Sent
Leslie Chan	● Rejected - Email Sent	Review Leslie Chan	Approve Reject	Reject Sent
annie	● Needs Review	Review annie	Approve Reject	
kim	● Accepted	Review kim	Approve Reject	Send Approve

The portal includes a search and filter bar at the top of the page with which I can use to search for or filter certain tutor applications. I can search by the applicant name or university and filter by application status, timezone, technology, and grade levels that applicant is willing to tutor. Below the search bar is a table that displays a single tutor application per a row. The information displayed in the columns for each tutor application is: 1) Name of applicant, 2) Application Status, 3) Review application button that displays a modal with all of the application information, 4) Approve and Reject buttons, and 5) Send Email button. I will explain the steps that an Executive team member would take to fully review a tutor application:

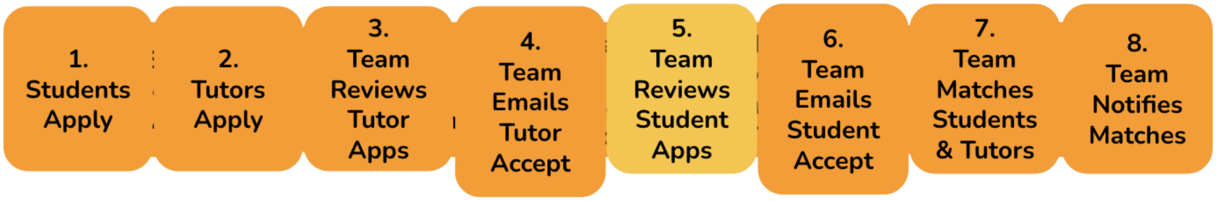
1. When an application is submitted by a prospective tutor, a new row will be shown in the table with a status of “Needs Review” (i.e. 4th row “annie” in picture)
2. I as an Executive member will click on the purple “Review” button which brings up a modal where I can see all of the application information for that applicant. I will read the answers and determine the eligibility of the applicant using our predetermined standards.
3. I decide that the applicant is a good fit for Hatch and want to approve the application. I click the green “Approve” button. Now a green “Send Approve” button shows up in the last column of the row. (i.e. 5th row “kim” in picture). At this point, I can decide to retract my approval and instead reject the applicant by pressing the red “Reject” button if I wish.
4. I decided to finalize my approval and I want to send an email to the applicant notifying them that we have accepted them as a Hatch tutor. I press the green “Send Approve” button and now all the buttons become disabled because no other changes can be made from the portal. (ie. 2nd row “imran”) This is the end of the review process for the applicant.

STEP 4: Team Emails Tutor Acceptances & Rejections

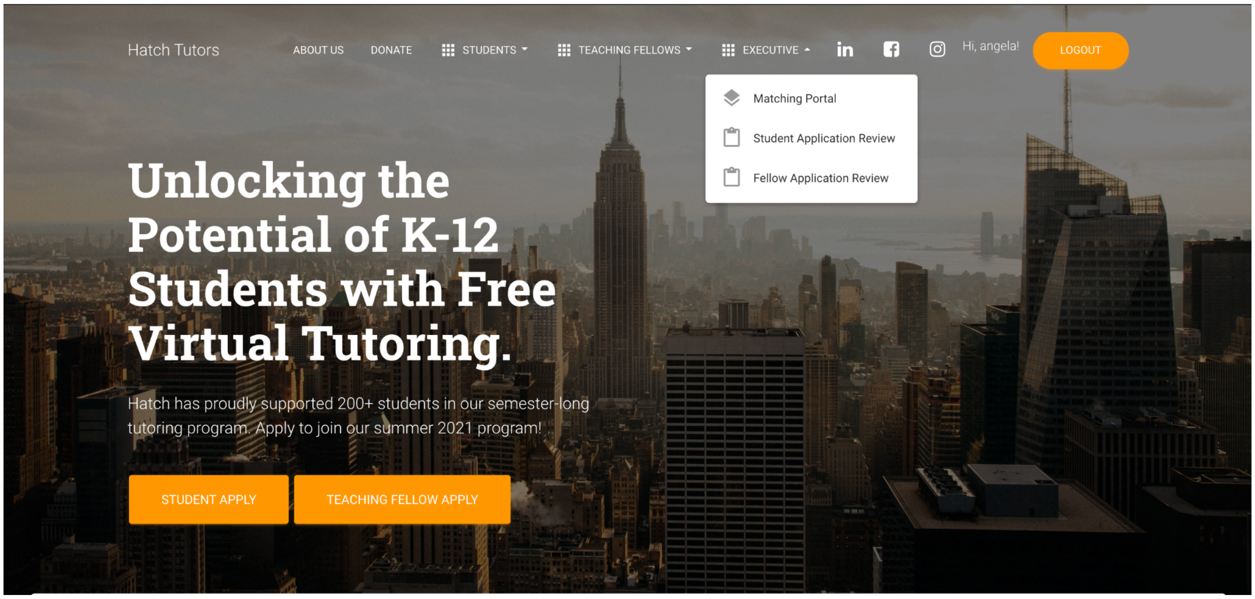

1. Students Apply
2. Tutors Apply
3. Team Reviews Tutor Apps
4. Team Emails Tutor Accept
5. Team Reviews Student Apps
6. Team Emails Student Accept
7. Team Matches Students & Tutors
8. Team Notifies Matches

This step was done manually in past program cycles by sending out individual emails to every applicant from our inbox. In the web application, the email notification is integrated within the application review portal. Executive team members can send the email by simply clicking the “Send Approve” button after approving the applicant, as explained above.

STEP 5: Team Reviews Student Applications



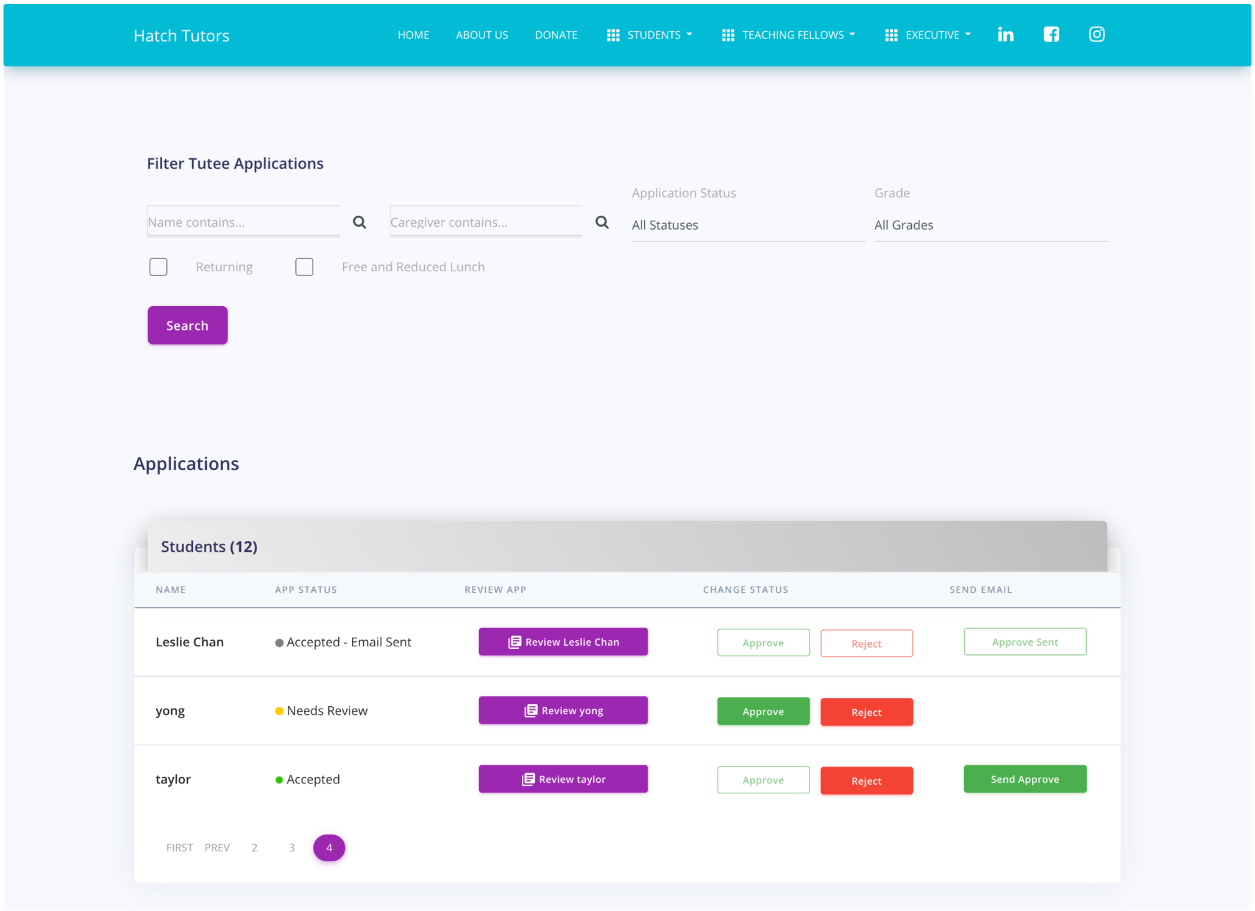
1. Students Apply
2. Tutors Apply
3. Team Reviews Tutor Apps
4. Team Emails Tutor Accept
5. Team Reviews Student Apps
6. Team Emails Student Accept
7. Team Matches Students & Tutors
8. Team Notifies Matches



Unlocking the Potential of K-12 Students with Free Virtual Tutoring.

Hatch has proudly supported 200+ students in our semester-long tutoring program. Apply to join our summer 2021 program!

I will walk through this section as a Hatch Executive team member who is reviewing student applications two weeks before the tutoring program begins. I click on “Student Application Review” from the “Executive” dropdown. This redirects me to a portal where I can review student applications.



Students (12)

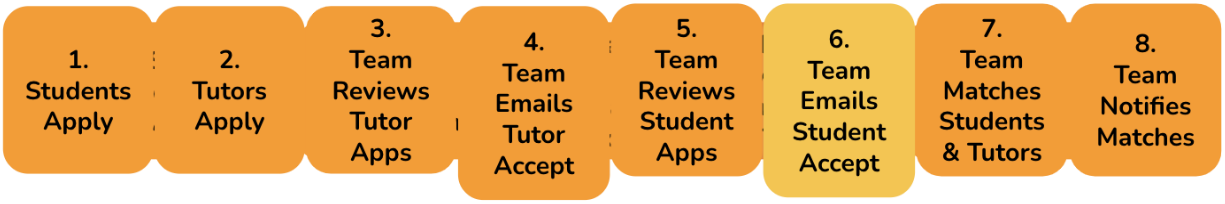
NAME	APP STATUS	REVIEW APP	CHANGE STATUS		SEND EMAIL
Leslie Chan	● Accepted - Email Sent	Review Leslie Chan	Approve	Reject	Approve Sent
yong	● Needs Review	Review yong	Approve	Reject	
taylor	● Accepted	Review taylor	Approve	Reject	Send Approve

The portal is very similar to the tutor application review portal. It includes a search and filter bar at the top of the page with which I can use to search for or filter certain student applications. I can search by the applicant name or caregiver name and filter by application status, grade level, returning students, qualification for free and reduced lunch which is a standard we use to determine eligibility. Below the search bar is a table that displays a single student application per a row. The information displayed in the columns for each student application is: 1) Name of applicant, 2) Application Status, 3) Review application button that displays a modal with all of the application information, 4) Approve and Reject buttons, and 5) Send Email button. I will explain the steps that an Executive team member would take to fully review a student application:

5. When an application is submitted by a prospective student, a new row will be shown in the table with a status of “Needs Review” (i.e. 2nd row “yong” in picture)

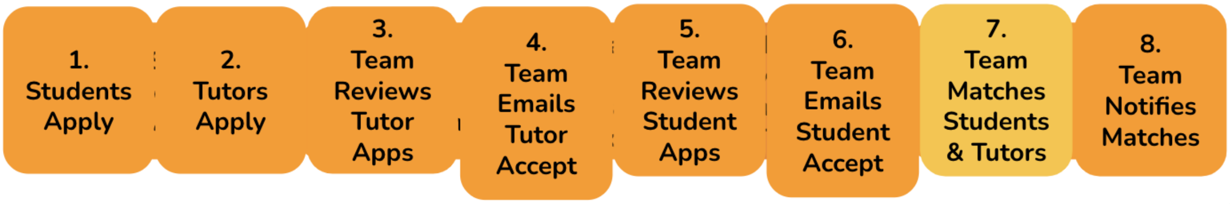
6. I as an Executive member will click on the purple “Review” button which brings up a modal where I can see all of the application information for that applicant. I will read the answers and determine the eligibility of the applicant using our predetermined standards.
7. I decide that the applicant is a good fit for Hatch and want to approve the application. I click the green “Approve” button. Now a green “Send Approve” button shows up in the last column of the row. (i.e. 3rd row “taylor” in picture). I can decide to retract my approval and instead reject the applicant by pressing the red “Reject” button if I wish.
8. I finalize my approval and I want to send an email to the applicant notifying them that we have accepted them as a Hatch student. I press the green “Send Approve” button and now all the buttons become disabled because no other changes can be made from the portal. (ie. 1st row “leslie”) This is the end of the review process for the applicant.

STEP 6: Team Emails Student Acceptances & Rejections

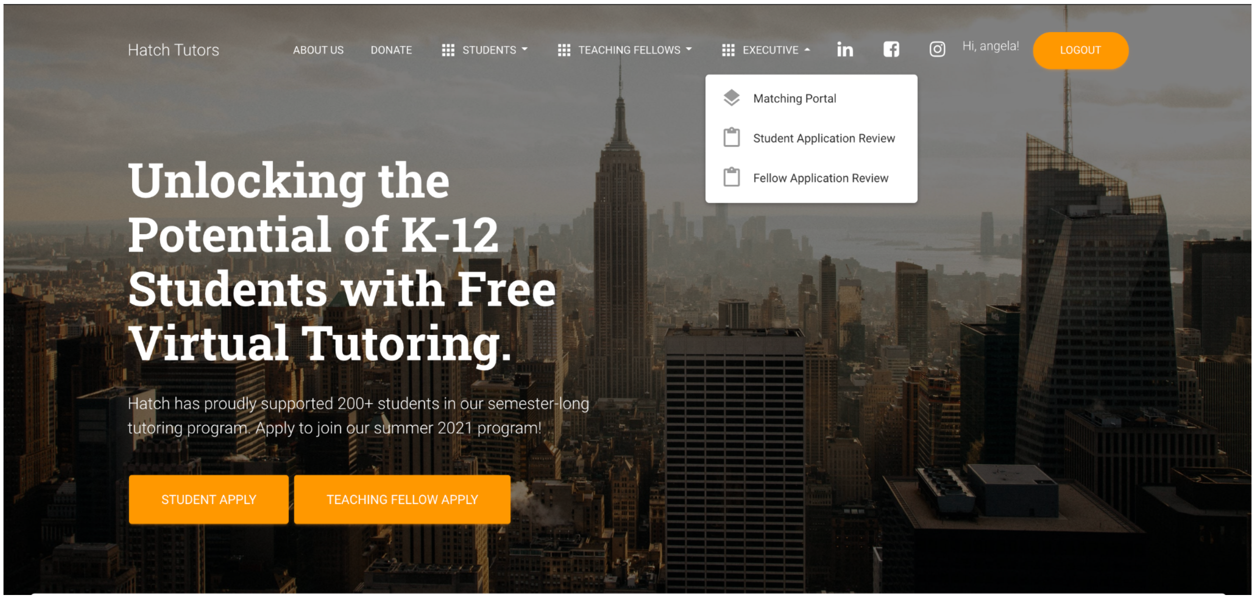

1. Students Apply
2. Tutors Apply
3. Team Reviews Tutor Apps
4. Team Emails Tutor Accept
5. Team Reviews Student Apps
6. Team Emails Student Accept
7. Team Matches Students & Tutors
8. Team Notifies Matches

As with tutors, this step was done manually in past program cycles by sending out individual emails to every applicant from our inbox. In the web application, executive team members can send the email by simply clicking the “Send Approve” button after approving the applicant, as explained above.

STEP 7: Team Matches Students & Tutors


1. Students Apply
2. Tutors Apply
3. Team Reviews Tutor Apps
4. Team Emails Tutor Accept
5. Team Reviews Student Apps
6. Team Emails Student Accept
7. Team Matches Students & Tutors
8. Team Notifies Matches

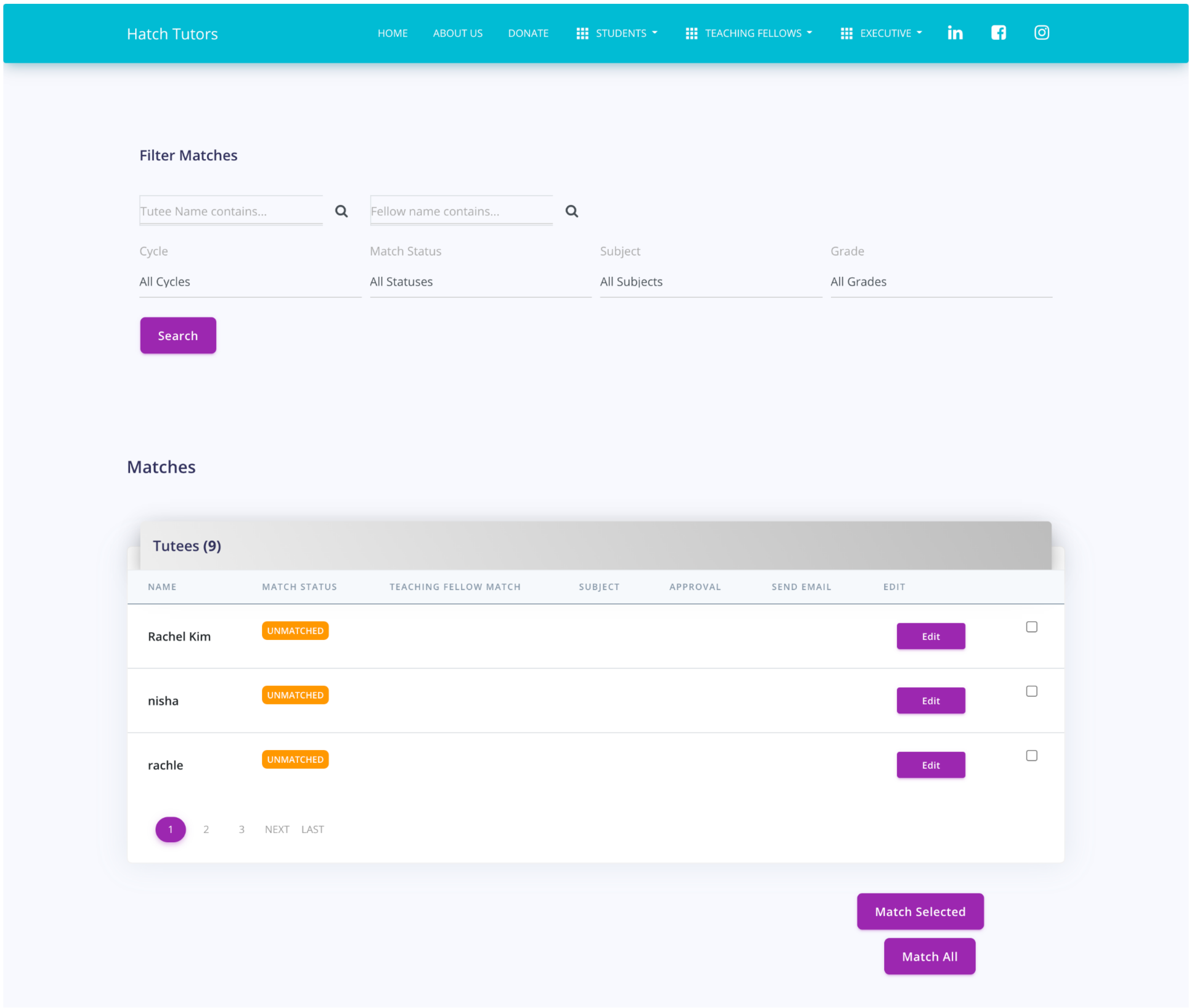
One week before the Hatch tutoring program begins, the Hatch Executive team will have made a decision to accept or reject all student and tutor applicants. Now, the Executive team needs to match students with tutors. To do this, I will assume the role of an Executive member again and navigate to the homepage and click on the “Matching Portal” from the “Executive” dropdown.



Unlocking the Potential of K-12 Students with Free Virtual Tutoring.

Hatch has proudly supported 200+ students in our semester-long tutoring program. Apply to join our summer 2021 program!

This brings me to the Matching Portal page:



Filter Matches

Tutee Name contains...

Fellow name contains...

Cycle	Match Status	Subject	Grade
All Cycles	All Statuses	All Subjects	All Grades

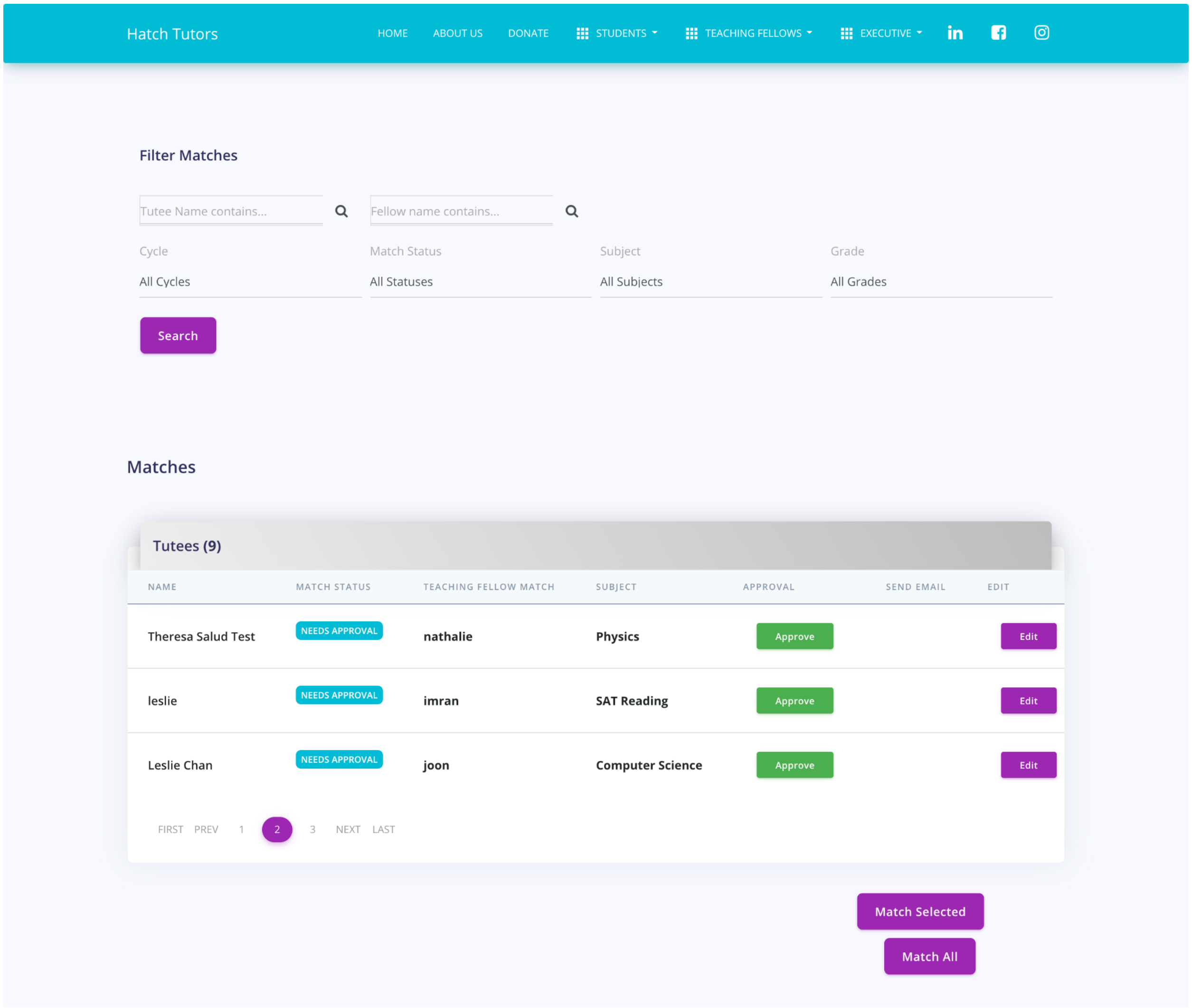
Matches

Tutees (9)

NAME	MATCH STATUS	TEACHING FELLOW MATCH	SUBJECT	APPROVAL	SEND EMAIL	EDIT	
Rachel Kim	UNMATCHED					Edit	□
nisha	UNMATCHED					Edit	□
rachle	UNMATCHED					Edit	□

There is a search bar and filter at the top of the page. I can use this to search for student names, tutor names, and filter by Program Cycle, Match status, Subject taught, and Grade level of the student. Below the search and filter is a table. Each row represents one student that has been accepted to the Hatch tutoring program. Per each row, the columns are 1) the student Name, 2) the Match Status, 3) the matched Tutor's name, 4) the subject to be tutored, 5) Match approval status, 6) Sending email notification about match, 7) manual match edit button. I will describe the process that Executive members will follow to create matches:

1. I will click the “Match All” button and this will run the matching algorithm in the back end that will pair students with tutors based on heuristics such as subject area, grade level of student, tutoring platform, and availability. The output of the matching algorithm will be rows in the table filled in with a matched tutor and subject. The status of the rows will be changed to “Needs Approval” and the green “Approve” button will appear.



Filter Matches

Tutee Name contains...

Fellow name contains...

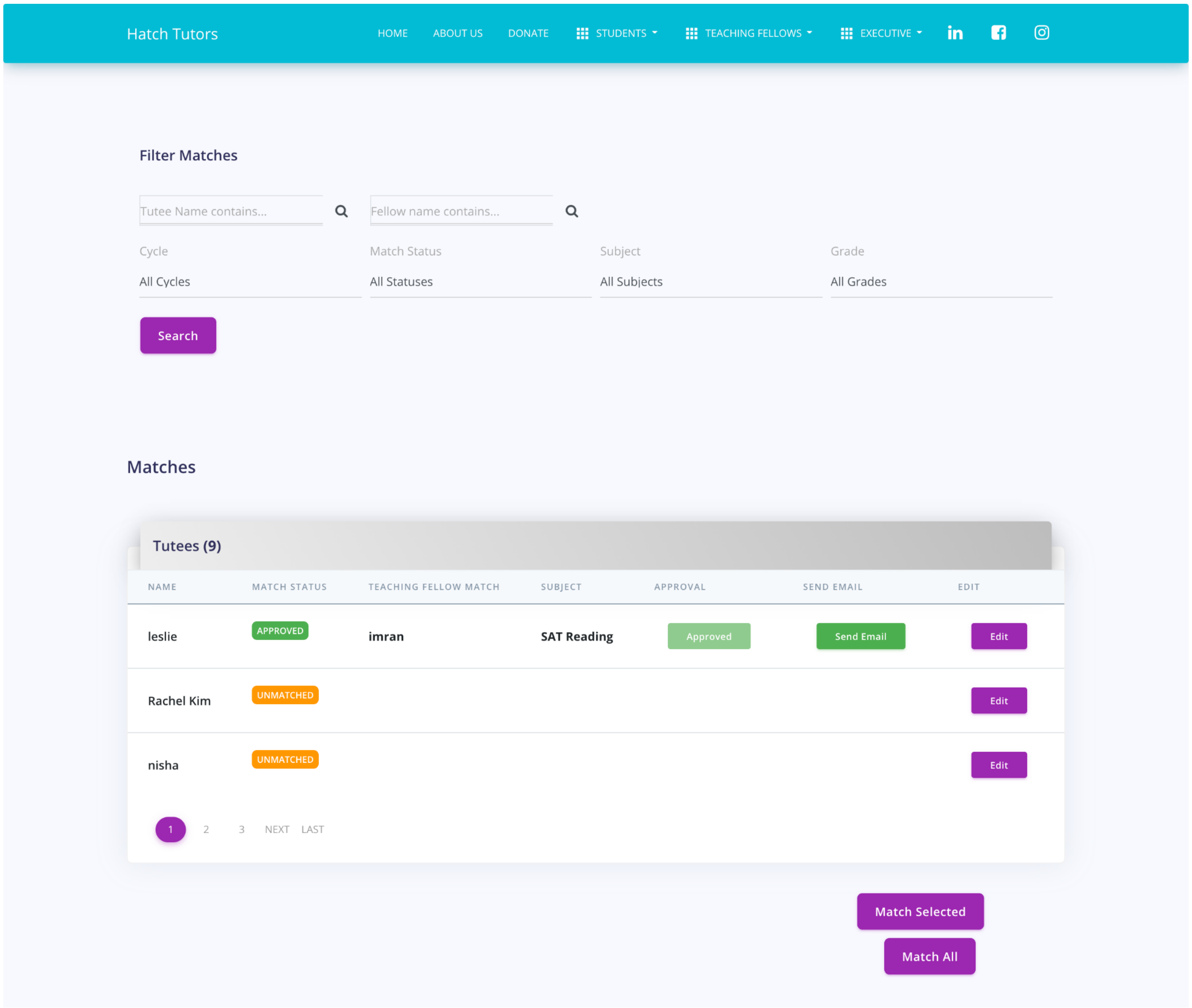
Cycle	Match Status	Subject	Grade
All Cycles	All Statuses	All Subjects	All Grades

Matches

Tutees (9)

NAME	MATCH STATUS	TEACHING FELLOW MATCH	SUBJECT	APPROVAL	SEND EMAIL	EDIT
Theresa Salud Test	NEEDS APPROVAL	nathalie	Physics	Approve		Edit
leslie	NEEDS APPROVAL	imran	SAT Reading	Approve		Edit
Leslie Chan	NEEDS APPROVAL	joon	Computer Science	Approve		Edit

2. I decide to approve the match that the algorithm outputs for the student named leslie. After pressing “Approve,” the “Send Email” button appears allowing me to send an email to leslie and imran notifying them of their match for the Hatch program.



Filter Matches

Tutee Name contains...

Fellow name contains...

Cycle: All Cycles

Match Status: All Statuses

Subject: All Subjects

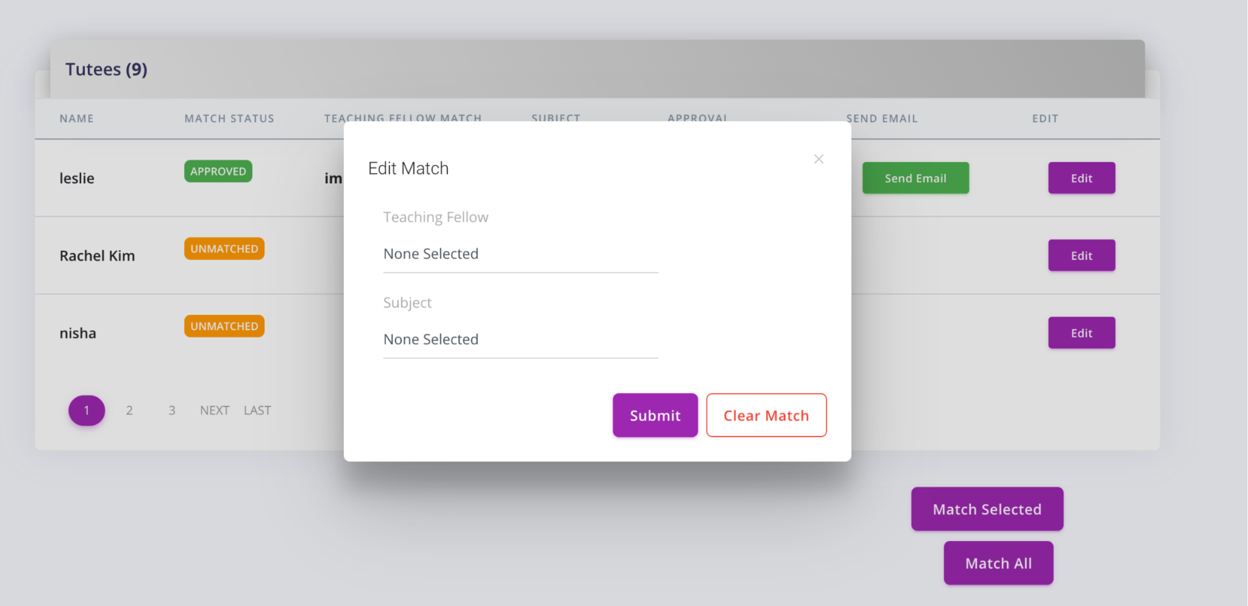
Grade: All Grades

Matches

Tutees (9)

NAME	MATCH STATUS	TEACHING FELLOW MATCH	SUBJECT	APPROVAL	SEND EMAIL	EDIT
leslie	APPROVED	imran	SAT Reading	Approved	Send Email	Edit
Rachel Kim	UNMATCHED					Edit
nisha	UNMATCHED					Edit

3. Sometimes the algorithm is unable to match a tutor to a student based on the programmed heuristics. Or, sometimes an Executive member wishes to manually edit a match output by the algorithm. Then, I as an Executive team member can press the purple “Edit” button and select a tutor and a subject for the student. I can also clear the match completely and the row will return to a state of “Unmatched.”



Tutees (9)

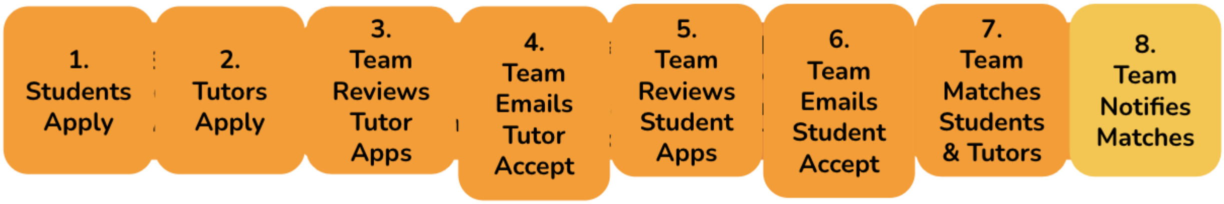
NAME	MATCH STATUS	TEACHING FELLOW MATCH	SUBJECT	APPROVAL	SEND EMAIL	EDIT
leslie	APPROVED	im			Send Email	Edit
Rachel Kim	UNMATCHED					Edit
nisha	UNMATCHED					Edit

Edit Match

Teaching Fellow: None Selected

Subject: None Selected

STEP 8: Team Matches Students & Tutors



1. Students Apply

2. Tutors Apply

3. Team Reviews Tutor Apps

4. Team Emails Tutor Accept

5. Team Reviews Student Apps

6. Team Emails Student Accept

7. Team Matches Students & Tutors

8. Team Notifies Matches

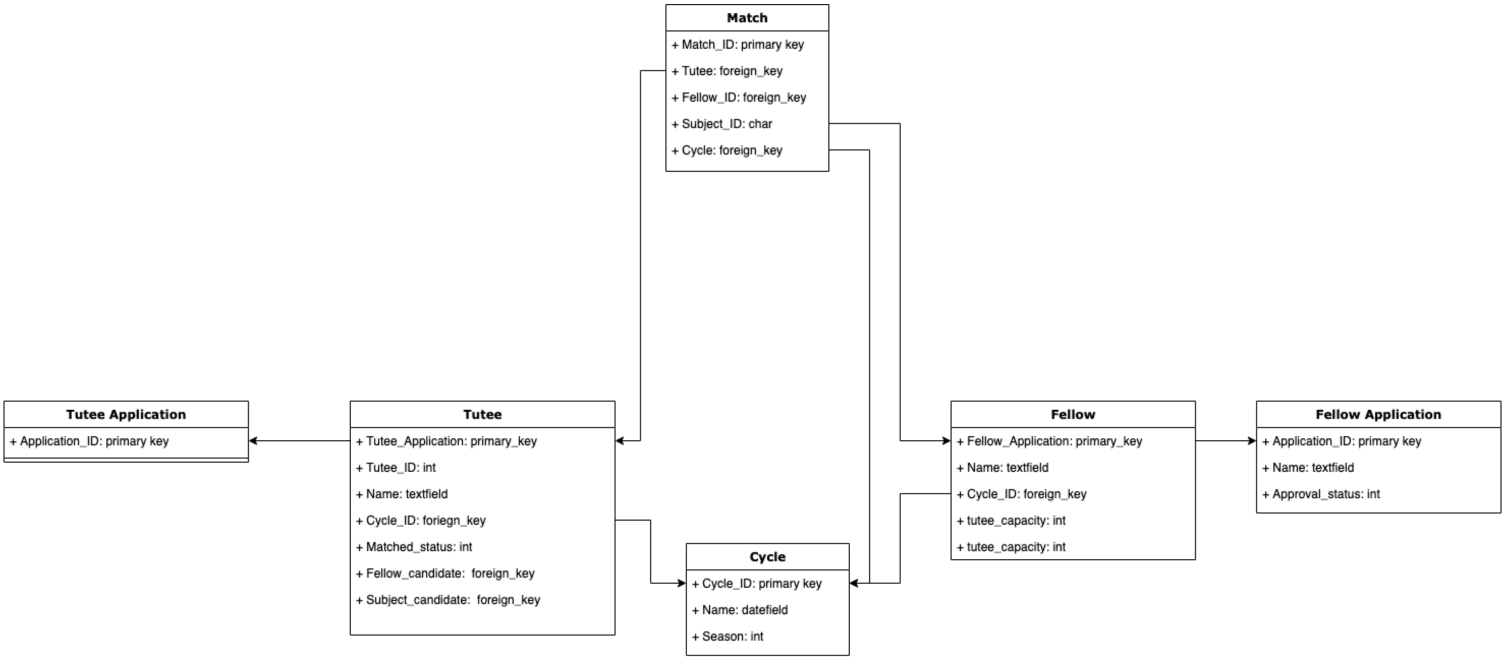
As with the application approval portals, the manual emailing has been converted to a “Send Email” button in the matching portal. Executive members can send an email notification to students and tutors about their pairing after approving the match, as previously shown.

4. Implementation: Design

In this section I will describe the design and code structure of the web application. I will discuss the database schema, the matching algorithm, the general structure of the code, and a few design challenges.

Database

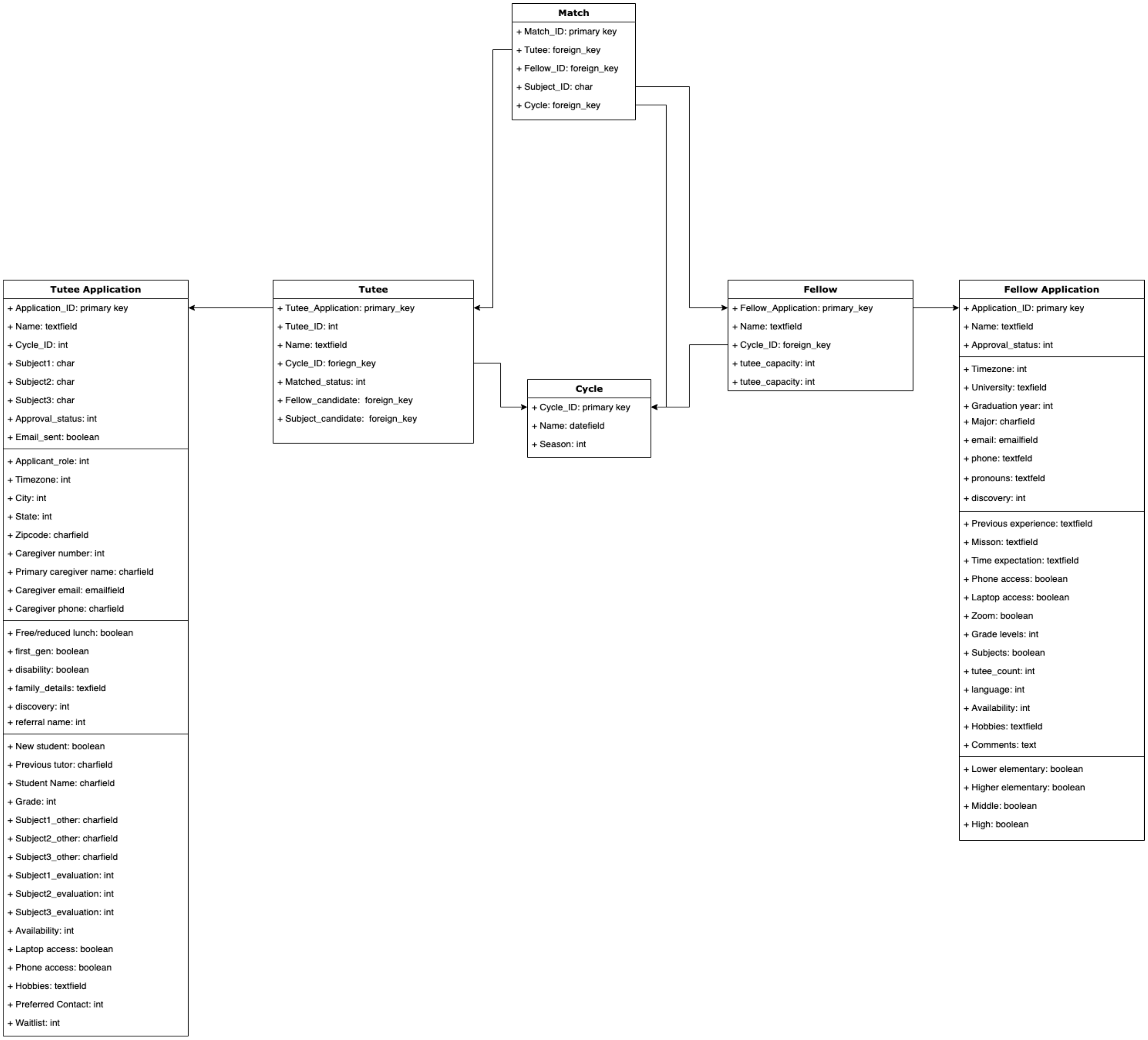
FIG: Database Truncated (ref. Next page for full tables)



This is a PostgreSQL relational database. The high-level relationship between the database tables are as such:

1. Tutee Application: represents a student application. Created upon student app submission.
2. Tutee: represents a student accepted into the Hatch program. Inherits from “Tutee Application.”
3. Fellow Application: represents a tutor application. Created upon tutor app submission.
4. Fellow: represents a tutor accepted into the Hatch program. Inherits from “Fellow Application.”
5. Match: represents a student-tutor pair in Hatch’s program. Match objects have a Tutee & Fellow.
6. Cycle: represents a tutoring program cycle. Tutee, Fellow, and Match objects have a cycle.

FIG: Database Full



The general structure of the database tables:

1. Tutee Application table

`Application_id =` primary key
`Approval_status =` Needs Approval, Approved, Rejected, Accept Email Sent, Reject Email Sent
`Name =` name of the student
`…(remaining fields correspond to student application answers)`

2. Tutee

`Tutee_application =` primary key. One-to-one relationship with Tutee Application and thus inherits all fields from Tutee App
`Tutee_ID =` Needs Approval, Approved, Rejected, Email Sent
`Name =` name of the student
`Cycle =` most current tutoring cycle of participation
`Matched_status =` Unmatched, Needs Approval, Approved, Email Sent
`TF_candidate =` potential tutor match (many-to-one relationship)
`Subject_candidate =` potential subject in tutoring

3. Fellow Application

`Application_id =` primary key
`Approval_status =` Needs Approval, Approved, Rejected, Accept Email Sent, Reject Email Sent
`Name =` name of the tutor applicant
…`(remaining fields correspond to student application answers)`

4. Fellow

`Fellow_application =` primary key. One-to-one relationship with Fellow Application and thus inherits all fields from Tutee App
`Name =` name of the tutor
`Cycle =` most current tutoring cycle of participation
`Tutee_capacity =` remaining capacity for students

5. Match

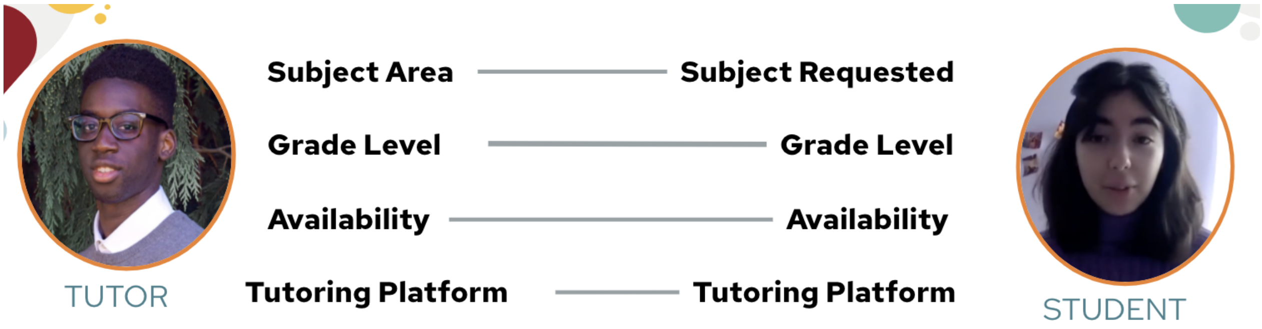
`Match_id =` primary key
`Cycle =` tutoring program cycle
`Tutee =` student
`Fellow =` tutor
`Subject =` subject tutored

6. Cycle

`Cycle_id =` primary key
`Name =` Name of the cycle
`Season =` Spring, Summer, Fall

Tutee Application and Fellow Application objects are created upon submission of the respective applications. Tutee and Fellow objects are created when an Executive member clicks on the "Approve" for an application. Match objects are created when an Executive approves a match. Cycle objects are currently created in the back end.

Matching Algorithm



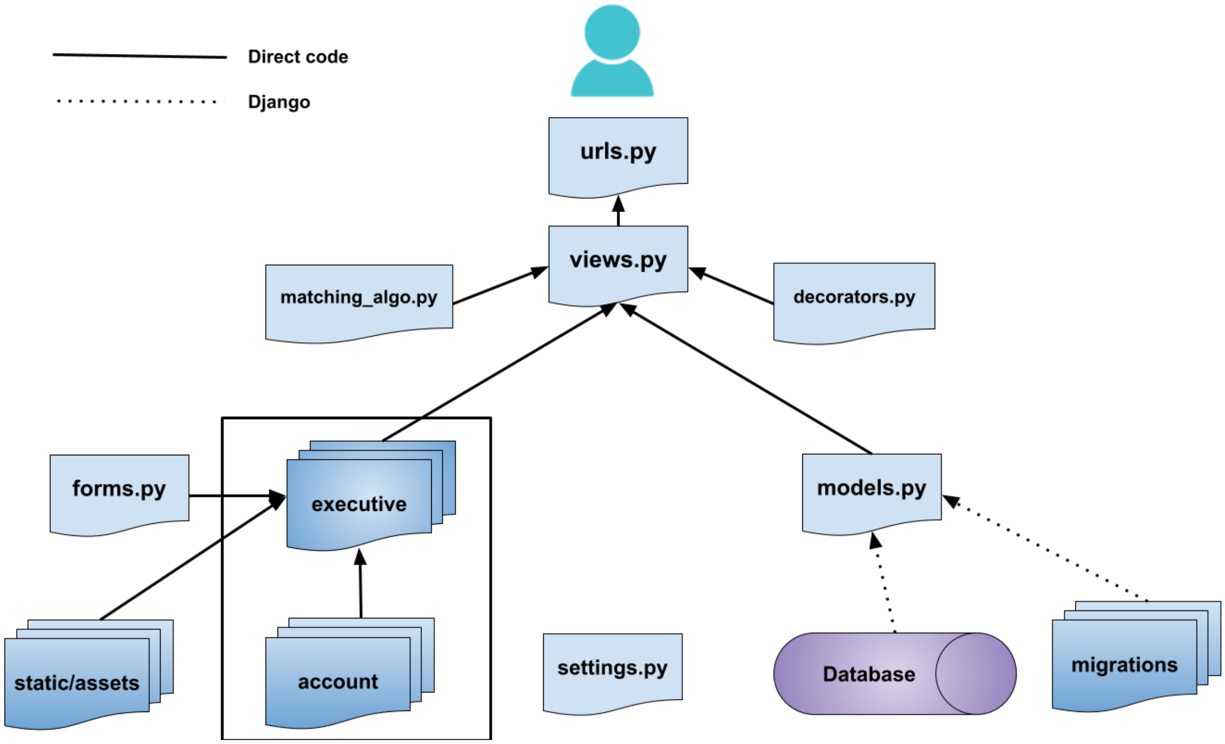
The matching algorithm is coded to represent the real-life procedure that Hatch uses to pair a student with a tutor:

1. A student can successfully be paired with a tutor if the following criteria is met 1) tutor is comfortable with requested subject area for tutoring, 2) tutor is comfortable tutoring grade level of student, 3) tutor and student align in at least one available time, 4) tutor and student align on access to tutoring platform (online laptop conferencing, phone conferencing, voice call). The algorithm obtains these criteria from the tutor and student application answers.
2. The algorithm will try to pair a student with a tutor that does not have any students before pairing with a tutor that has previously been assigned a student.
3. The algorithm will attempt to pair students with lower confidence in their subject area with tutors that have highest tutoring capacity. (i.e. all else equal in (1) and (2), if Tutor Rachel has capacity for 3 students and Tutor Nathalie has capacity for 2 students, Rachel will get matched with a student that has lower confidence in their subject)
4. The algorithm attempts to match students with their 1st choice subject first, then their 2nd choice, and lastly their 3rd choice.
5. The algorithm attempts to pair students first who want tutoring in the subject with the least amount of tutors available. (i.e. if 3 tutors can do middle school math and 10 tutors

can do high school English, then the algorithm will pair students who need tutoring in middle school math first.)

I implemented the algorithm using a python script and wrote tests using tutor and student application data imported from a CSV file. The code is available at: https://github.com/Angelali98/hatch-matching. After testing the python script, I implemented the algorithm in the web application by following the same structure with modifications. For example, instead of using a python dictionary to hold application data as I did in the script, I used Django QuerySet to filter data from the Tutee table and Fellow table in the database.

Structure of Code



Github: https://github.com/Angelali98/hatch

I will describe the structure of the code with reference to this diagram. The blue gradient icons are folders. The plain blue icons are files. The icon at the top is the user. The legend shows a solid black line which says “Direct code.” This means that files or folders connected by a solid black arrow directly references code. For example, there is a solid black line from `matching_algo.py` to `views.py`. This is because view.py directly calls a function written in `matching_algo.py`. The dashed arrow refers to files that the Django framework automatically connects. The box surrounding `executive` and `account` represents their parent folder called `templates.`

If you open the Github link you will find:

1. Generated Files: `requirements.txt` (list of dependencies), `manage.py` (Django command-line utility), `Procfile` and `.DS_Store` (configuration for OS and Heroku).
2. Folder: `hatch.`
    a. `settings.py –` auto-generated Django file I modified with installment settings for Postgres, Heroku hosting, authentication, and the main application.
3. Folder: `executive –` code for the application. All of the files and folders in the diagram except `settings.py` are in this folder. (Not to be confused with the sub-folder shown in the diagram also called `executive)`
    a. `models.py – Model` is a class that represents database tables. Each attribute / field is a table column
    b. `migrations –` django generated records of changes to models
    c. `static/assets –` holds the images used in the application
    d. `account –` imported code from Django allauth package for Google authentication. I modified some of the base templates to fit the application aesthetic.
    e. `forms.py –` back end code for the student and tutor application. The class `ModelForm` converts the models (database tables) TuteeApplication & FellowApplication into a Django form object.

f. `executive` - folder containing all the HTML templates I created for this application. The templates include Django template language for looping and accessing database objects. The notable HTML templates are:

   i. `landing-page.html` (homepage)

   ii. `tuteeapp.html` (student application),

   `tuteeapp_confirmation.html` (app submission confirmation)

   iii. `applications.html` (student app review portal),

   iv. `fellowapp.html` (tutor application),

   `fellowapp_confirmation.html` (app submission confirmation)

   v. `fellow_applications_review.html` (tutor app review portal)

   vi. `matching.html` (matching portal)

g. `matching_algo.py` - code for the matching algorithm. Includes a function `clearMatches` that will clear the matches given Tutee objects

h. `decorators.py` - used to restrict executive portals to certain users

i. `views.py` - a view function takes a Web request and returns the HTML response. Filtering and making updates to database objects happens here, along with email sending and table pagination.

j. `urls.py` - maps view functions to URLs that the user can access

Approximate metrics on the code I implemented:

- Lines of HTML: 4300
- Lines of python
  - App: 1520
  - Matching algorithm script: 800
- Lines of code total: 6620

Design Challenges

1. The database required careful design. The tables must be set up properly in consideration for how the web application might expand to accommodate the future expansion of the Hatch program. One major decision I made was to make a Tutee table separate from the Tutee Application table and same for Fellow and Fellow Application. This is because students and tutors may return to the Hatch program multiple times and their information may change. Having a separate Tutee object means that the original application data can remain unaltered. In addition, the Tutee object can inherit fields from the Application object to reduce redundancy, and I can also add fields to the Tutee model in the future to inherit information from a “Returning tutee” application if need be. Similarly, I decided to make a separate Match table instead of just having a “matches” field on Tutees and Fellows because this will allow for easier tracking of analytics and metrics, which is important information for Hatch.
2. Designing the matching algorithm was challenging at the start because I had to choose the proper python data structures to use for storing information from the CSV files in the python script. Dictionaries and lists were very handy choices because they allowed me to map a student name to a list of information. Python was a good choice because its syntax allows for easy manipulation of tuples, list, and dictionaries. When I integrated the algorithm into the web application, it was actually significantly easier to implement than in the script. This is because Django QuerySet allows for filtering functionality and the Tutee database objects allow me to directly access information about a student from the object fields. This eliminated the necessity for data structures such as dictionaries.
3. I also ran into an implementation challenge with the Application Review Modal in the Student application review portal. The modal was showing one student’s application information for every row instead of showing the correct application information for each

different student. With the help of my advisor, I was able to figure out that this was a naming problem within the HTML that displayed the table. Django template language allows me to iterate through a list of database objects (Tutee Applications) in an HTML template. For each iteration, I implemented code to display a table row including the Review modal. The modal must have a unique name referring to each student's application object.

5. Evaluation

I conducted evaluations with users from the beginning of the development process to the end.

1. I interviewed Hatch executive members to help iterate on the hi-fi inVision mockups
2. I conducted a total of 5 user interviews evaluating the web application in production. I used the task list in APPENDIX 2 to guide the users through the application. Feedback gathered from the interviews is in APPENDIX 3. Here is a summary:
    - Midpoint user interview with my advisor Robert Dondero (no task list, evaluated up through application review portals)
        - It was useful to see a user who is not intimately familiar with the review process try the application. The user seemed to smoothly understand the flow of the application and the review portal usage. Constructive feedback pertained to issues such as state of the system alerts (submission confirm etc.), form logic, and some UI changes.
    - 2 interviews with Hatch Executive team members in charge of application review
        - These two interviews were with Hatch members that work with the application review process. They also found the approval flow intuitive and mainly had future scope suggestions for the review portals such as ability to add notes, initials, and email template additions.

- 2 interviews with Hatch team members: UI/UX engineer and a software engineer
  - These individuals also work with the review process. One was very detailed with testing edge cases and managed to catch a bug in the matching algorithm that I then fixed. They also had specific future scope suggestions such as a static single source of truth for matches since the matching portal is editable and not well-fitted for this purpose. The UI engineer also had suggestions about leveraging icons for clarity.

3. I self-evaluated the application using Jakob Nielsen's UI Heuristics.[3] Here is a summary of my evaluation:
   - **Visibility of system status:**
     - The student and tutor applications both contain sections with descriptive headers for the user. The system alerts the user with a "submission confirmation" page after a user submits an application.
     - In the Student Approval, Fellow Approval, and matching portals the filter in the table header alerts the user to how many rows were filtered. The buttons in the table become disabled after they are pressed. The "App Status" column alerts the user of the application status.
   - **Match between system and the real world:**
     - The application review and matching portals are a 1-to-1 mapping of the real-world process that Hatch uses. The terminology in these portals, such as "Tutee" and "Fellow" and the filter and status terms, is language used by Hatch.
     - The homepage uses language familiar to general users. It says "student" in lieu of "Tutee" and it explains what "Teaching Fellow" is upfront.

[3] https://www.nngroup.com/articles/ten-usability-heuristics/

- **User control and freedom**
    - The application review portals allow the user to freely switch between the reject and approval button until the email is sent. The user should not be allowed to make changes in the portal after the email is sent. The ”Review” modal has a clear close button.
    - The matching portal contains an “Edit” button for every row which allows the user to freely edit the match and clear the match.
- **Consistency and standards**
    - The navigation bar and social media icons are consistent throughout the web application.
    - The call to action buttons are consistent colors - redirections are orange, submits are purple, approvals are green, rejections are red.
    - The portals are consistent in their status terminology and table flow.
- **Error prevention**
    - The student and fellow applications contain form validation.
        1. * must be filled in
        2. Used dropdowns if possible to eliminate free-form text
        3. Email validation, numerical range validation
    - Review and matching portals require a separate “Send email” action after approval and rejection. Send email does not appear until an approval or rejection is made. A good future scope addition would be to require two member signatures before allowing the Send Email button. Undo actions are supported: approval, rejection, and clear match options. After email is sent the user is no longer able to make any edits so the buttons are disabled.
- **Recognition rather than recall**
    - Buttons and columns are labeled in every table in the application portals

    - The application review button in the approval portals allow the user to look at application information
    - It would be worthwhile to see if Executive members can benefit from having an application Review modal for rows in the Matching portal
- **Flexibility and efficiency of use**
    - The application review and matching portals allow for flexibility using the filters and the ability to edit each row. It would be beneficial to add bulk actions for efficiency such as Approve all, Send Email to all approved, etc.
- **Aesthetic and minimalist design**
    - I leveraged a very well-made free Bootstrap 4 template for minimalist design: Material Kit
- **Help Users Recognize, Diagnose, and Recover from Errors**
    - The application form validation will alert the user with a description of an error if the user tries to submit with an invalid answer.
    - The approval and matching portals have many error prevention tactics as discussed previously. There are no error alerts because the approval process depends on the human intention and the system cannot know if an “Approval” was an error.
- **Help and Documentation**
    - There currently is not much documentation in this application as it was not within the scope of the independent work. I will soon be implementing pages that describe the Hatch program and eligibility criteria. Another feature that may help Executive members is documentation on how to use the portals. Perhaps at the top of the table columns, there can be a popup that briefly describes each column or perhaps a separate page with a step-by-step might be helpful. I will test this with the team.

6. Conclusion and Future Work

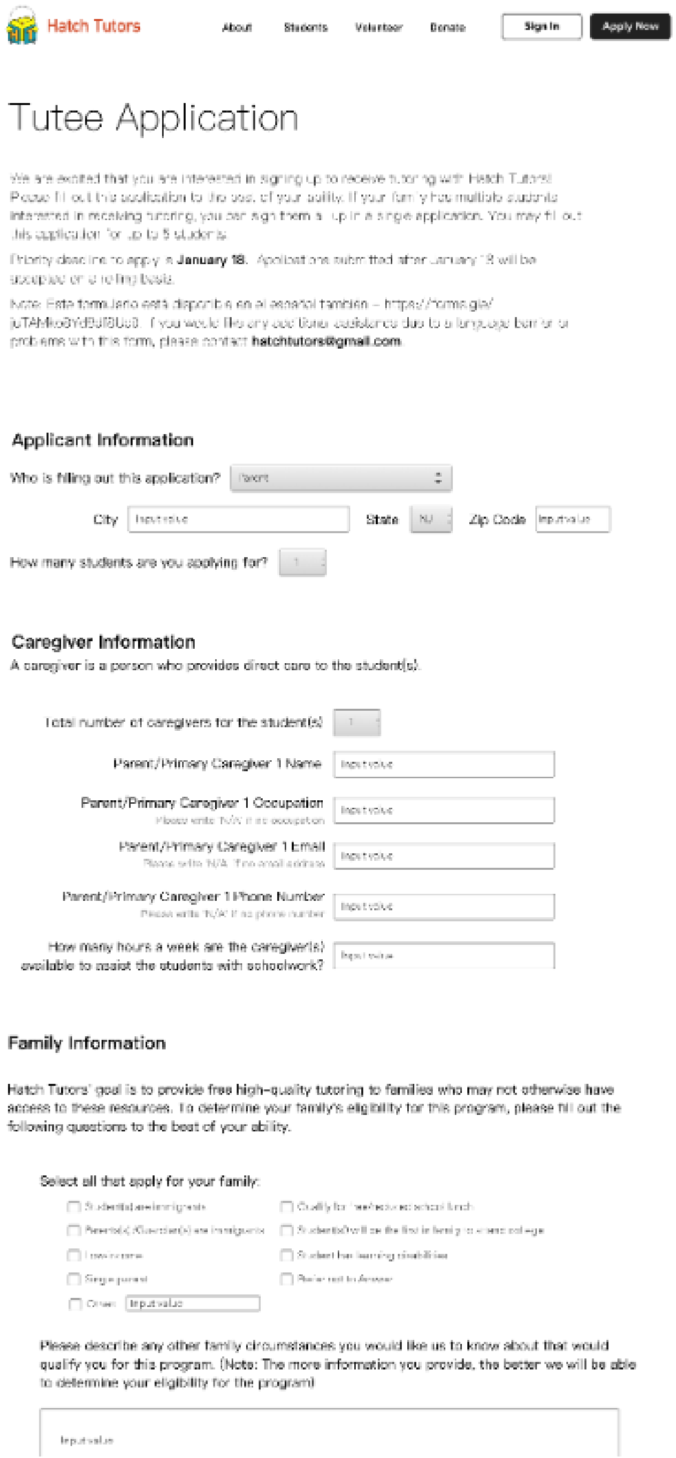
In conclusion, the Hatch Tutors web application is designed to now replace and automate the tedious application review and matching process for the Hatch Executive team. This will be of tremendous benefit in alleviating the bottleneck for scaling the tutoring program and I look forward to using the web application in production. The user interviews I conducted suggest that the design of the web application is an intuitive mapping of the manual review process. The basic functionality is in place, and the work that must be done before we use the application in production is: designing the Hatch student and tutor program description pages and other informational pages such as About Us and Testimonials, finalizing text and visuals on the landing page, finalize email templates, and a thorough web-security testing potentially with outside help. I plan to put the web application in public use by the fall of 2021 and recruit other volunteers to collaborate with in further building the application. There are many exciting opportunities for future work, such as a portal for tutors to track sessions and resources or advanced matching, that will benefit the Hatch Tutors organization!

7. Acknowledgements

I would like to thank all the people who have supported me throughout the completion of this Independence Work project. First, I would like to thank my advisor Prof. Robert Dondero for providing me with guidance weekly over Zoom from pre-project planning to evaluation. I would like to thank Theresa Salud, my Hatch colleague and UI/UX engineer for helping me learn the design process. I would also like to thank the Hatch team members for volunteering their time with me in building the Hatch organization, defining the review process, and helping me with the web application user interviews. I also want to acknowledge the individuals whose online resources I referenced in building my web application. In particular, I give credit to Creative Tim for creating Material Kit, the free Bootstrap template that I leveraged for my web application aesthetics. I also would like to acknowledge StackOverflow, Dennis Ivy Django tutorials on YouTube, all the other resources I have referenced to build my project. Lastly, I would like to thank the Princeton Independent Work Program.

8. Appendices

Appendix 1: inVision Mockups



Tutee Application

We are excited that you are interested in signing up to receive tutoring with Hatch Tutors! Please fill out this application to the best of your ability. If your family has multiple students interested in receiving tutoring, you can sign them all up in a single application. You may fill out this application for up to 5 students.

Priority deadline to apply is **January 18.** Applications submitted after January 18 will be accepted on a rolling basis.

Note: Este formulario está disponible en el español también – https://forms.gle/juTAMko3YdBUBUs9. If you would like any additional assistance due to a language barrier or problems with this form, please contact **hatchtutors@gmail.com**.

Applicant Information

Who is filling out this application? Parent

City Input value State NJ Zip Code Input value

How many students are you applying for? 1

Caregiver Information

A caregiver is a person who provides direct care to the student(s).

Total number of caregivers for the student(s) 1

Parent/Primary Caregiver 1 Name Input value

Parent/Primary Caregiver 1 Occupation Input value
Please write N/A if no occupation

Parent/Primary Caregiver 1 Email Input value
Please write N/A if no email address

Parent/Primary Caregiver 1 Phone Number Input value
Please write N/A if no phone number

How many hours a week are the caregiver(s) available to assist the students with schoolwork? Input value

Family Information

Hatch Tutors' goal is to provide free high-quality tutoring to families who may not otherwise have access to these resources. To determine your family's eligibility for this program, please fill out the following questions to the best of your ability.

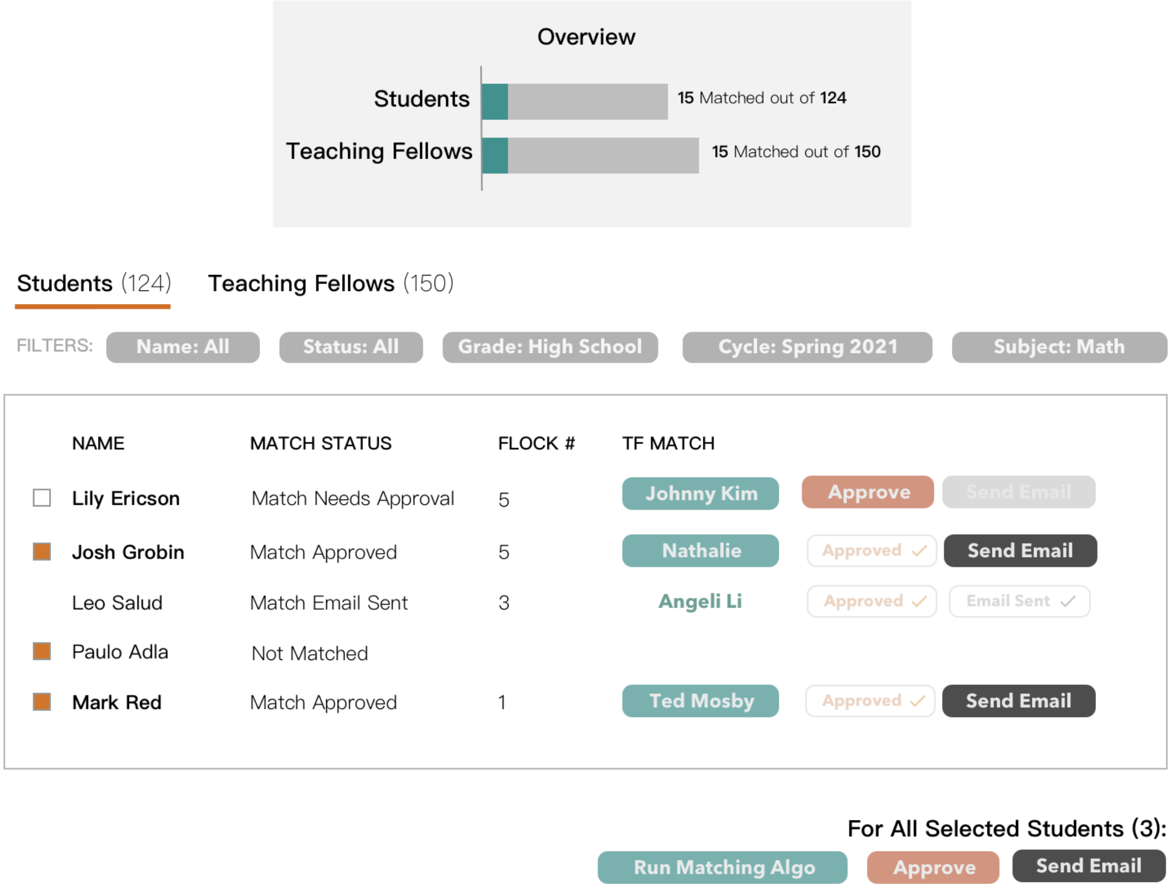
Select all that apply for your family:

- □ Student(s) are immigrants
- □ Parent(s)/Guardian(s) are immigrants
- □ Low income
- □ Single parent
- □ Other: Input value
- □ Qualify for free/reduced school lunch
- □ Student(s) will be the first in family to attend college
- □ Student has learning disabilities
- □ Prefer not to Answer

Please describe any other family circumstances you would like us to know about that would qualify you for this program. (Note: The more information you provide, the better we will be able to determine your eligibility for the program)

Input value

Matching Tool



Overview

Students: 15 Matched out of 124

Teaching Fellows: 15 Matched out of 150

Students (124) **Teaching Fellows** (150)

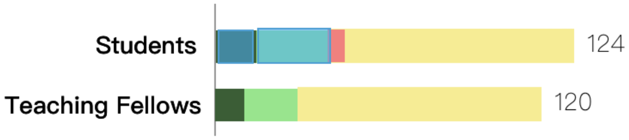
FILTERS: Name: All | Status: All | Grade: High School | Cycle: Spring 2021 | Subject: Math

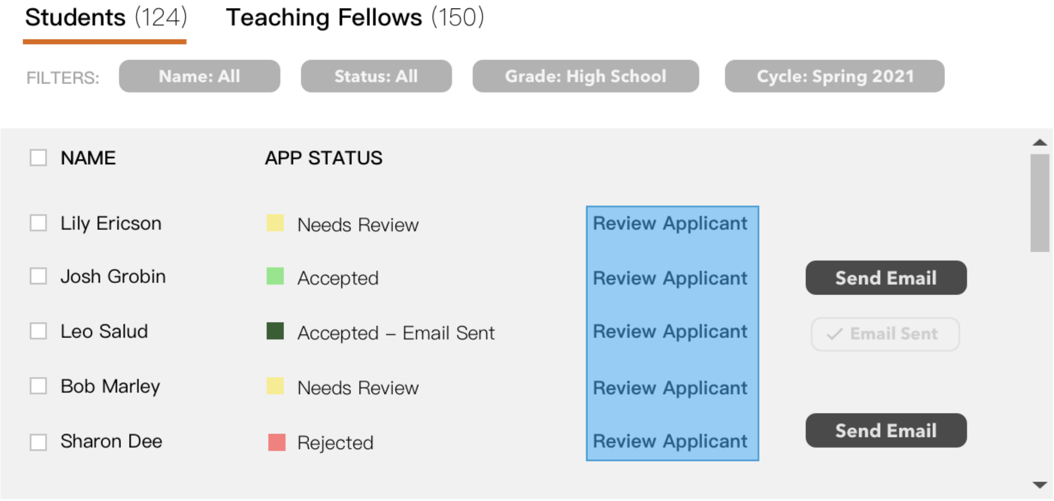
	NAME	MATCH STATUS	FLOCK #	TF MATCH		
□	Lily Ericson	Match Needs Approval	5	Johnny Kim	Approve	Send Email
☑	Josh Grobin	Match Approved	5	Nathalie	Approved ✓	Send Email
	Leo Salud	Match Email Sent	3	Angeli Li	Approved ✓	Email Sent ✓
☑	Paulo Adla	Not Matched				
☑	Mark Red	Match Approved	1	Ted Mosby	Approved ✓	Send Email

For All Selected Students (3):

Run Matching Algo | Approve | Send Email

Applications

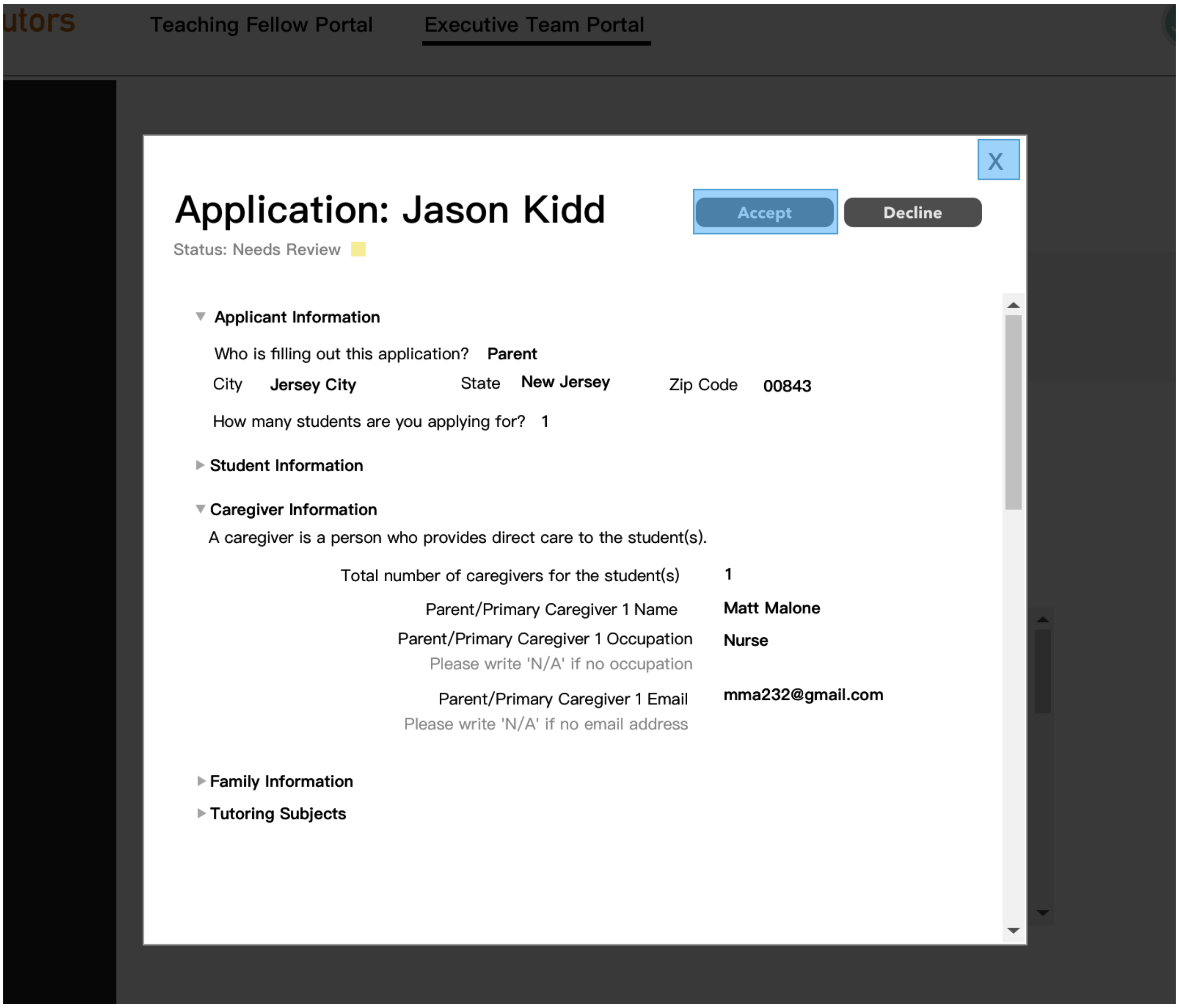




Students (124)   Teaching Fellows (150)

FILTERS: Name: All   Status: All   Grade: High School   Cycle: Spring 2021

□ NAME	APP STATUS		
□ Lily Ericson	🟨 Needs Review	Review Applicant	
□ Josh Grobin	🟩 Accepted	Review Applicant	Send Email
□ Leo Salud	🟩 Accepted – Email Sent	Review Applicant	✓ Email Sent
□ Bob Marley	🟨 Needs Review	Review Applicant	
□ Sharon Dee	🟥 Rejected	Review Applicant	Send Email



Application: Jason Kidd

Status: Needs Review

Applicant Information

Who is filling out this application? Parent

City Jersey City State New Jersey Zip Code 00843

How many students are you applying for? 1

Student Information

Caregiver Information

A caregiver is a person who provides direct care to the student(s).

Total number of caregivers for the student(s) 1

Parent/Primary Caregiver 1 Name Matt Malone

Parent/Primary Caregiver 1 Occupation Nurse

Please write 'N/A' if no occupation

Parent/Primary Caregiver 1 Email mma232@gmail.com

Please write 'N/A' if no email address

Family Information

Tutoring Subjects

Appendix 2: User Interview Task List

You are a Student

1. Learn about the Hatch community.
2. Apply to be a tutee. Check free-and-reduced lunch

You an executive member

1. Login
2. Check if tutee is in tutee app review portal

3. Review the tutee application
4. Accept all free-and-reduced lunch applications
5. Reject Johnny Kim (improper application)
6. Send acceptance & rejection emails to applicants

Student

1. Check in your email for acceptance or rejection email

You are an aspiring Teaching fellow

1. Find out about some of the current teaching fellows.
2. Apply to be a fellow.

You an executive member

7. Check if fellow is in fellow app review portal
8. Review the fellow application and accept or reject the application
9. Send email to an accepted applicant
10. Accept all Princeton university applications

11. Run the matching algorithm
12. Manually assign a teaching fellow and subject to a tutee
13. Approve the match and notify them by email of the match

Appendix 3: User Interview Task List

Homepage

Prof. Dondero

- Hide login except for executive team members
- Apply summer fellow button
- Change text to "Meet some of our fellows!"
- Take off links that don't work or instead just put "page coming soon"

Leslie, Rachel

- Meet our teaching fellow - have a brief description of what teaching fellow is
- Have a display where the metrics goes up
- Footer with contact

Tutee Application

Prof. Dondero

- Have a confirmation page for after submitting an application

- Darken the text
- Dynamic form for the returning students also the tutoring subjects
- Check that the navbar links are correct
- Form validation
- Explain asterisk
- Explain selection criteria
- More indicative dropdown

Leslie, Rachel

- Check spelling

T, Cathleen

- Future Scope: Application for Students is a tad long if goal is to increase applications and make it as accessible as possible
  - maybe split up into sections to make processes more engaging with progress bar
- Suggestion: confirmation email. “Here’s your application, thanks applying, etc”

Fellow Application

Prof. Dondero

- Have a confirmation page for after submitting an application
- Darken the text
- Form validation
- Explain asterisk
- More indicative dropdown

Leslie, Rachel

- FELLOW app -- have returning app

Student Application Review Portal

Prof. Dondero

- Badge → change to colored text
- Sustain search query
- Filter indication

Leslie, Rachel

- Add the onboarding survey to the email send out
- Email waivers for families to fill out
- People take notes on portal on each application
- add information for executive member initials after reviewed
- Change some of the form inputs

- “Caregiver NAME contains”
- Tutee and fellow review -- send all email to reject and accept

T, Cathleen

- Suggestion: Student Application review page - date of application should be here, order by last edited
- Suggestion: Colors on page should be changed. Perhaps use of icons.
- Suggestion: There should be a free and reduced Lunch column perhaps
- **Future: Ability to edit application information and application flow for returning TFs and Students who are already in our database**

Fellow Application Review Portal

Prof. Dondero

- Badge → change to colored text
- Sustain search query

Leslie, Rachel

- people take notes on portal on each application
- add information for executive member initials after reviewed

Same as Student review portal

Matching Portal

Prof. Dondero

- Badge → change to colored text
- Ability to clear all matches
- Unclear whether it matches on the filtered or not filtered part
- Sustain search query

T, Cathleen

- **Future: source of truth for finalized match information**
- **Check spelling**
- **Overall info button with usage details**